

ŘADA B - PRO KONSTRUKTÉRY

ČASOPIS PRO ELEKTRONIKU
ROČNÍK XLI/1992 ● ● ČÍSLO 2

V TOMTO SEŠITĚ

INTEL se představuje 41

MĚŘENÍ ELEKTRICKÝCH VELIČIN
NA POČÍTAČI PC

Realizace převodníků	43
Nejednodušší převodník D/A	43
Nejednodušší převodník A/D	43
Převodníky A/D a D/A	
na „univerzální“ desce s plošnými spoji	46
3kanálový 8bitový převodník A/D	46
3kanálový 8bitový převodník A/D	53
Jednokanálový 8bitový převodník A/D	54
Nejednodušší převodník A/D na desce s plošnými spoji	55
3kanálový 8bitový převodník A/D s automatickým přepínáním rozsahu	59
Nářmy pro použití desky převodníků	59
Zkušenosti z provozu a možnosti úpravy	60
 Dodatek (3kanálový 12bitový převodník A/D a 8bitový výkonový převodník D/A)	61
Programovatelné generátory s napětí	65
Programové vybavení	68
 JEDNOČIPOVÉ MIKROPOČÍTAČE	
Rozbor architektury mikropočítačů řady 8051	71
Přehled mikropočítačů řady 8051	71
Organizace paměti	72
Časování CPU	72
Struktura kanálů V/V	73
Čítače/časovače	75
 INZERCE	80

AMATÉRSKÉ RADIO ŘADA B

Vydavatel: Vydavatelství MAGNET-PRESS, s. p. 135 66 Praha 1, Vladislavova 26, tel. 26 06 51.

Redakce: 113 66 Praha 1, Jungmannova 24, tel. 26 06 51. Šéfredaktor L. Kalousek, OK1FAC, linka 354, sekretariát linka 355.

Tiskne: Naše vojsko, tiskárna, závod 08, 160 05 Praha 6, Vlastina ulice č. 889/23.

Rozšířuje: Poštovní novinová služba a vydavatelství MAGNET-PRESS s. p., Objednávky příjma: každá administrace PNS, pošta, doručovatel a předplatitelská střediska a administrace vydavatelství MAGNET-PRESS s. p., 113 66 Praha 1, Vladislavova 26, tel. 26 06 51-9. Poletní předplatné 29,40 Kčs. Objednávky do zahraničí vyfizuje ARTIA, a. s., Ve směrcích 30, 111 27 Praha 1.

Inzerci přijímá osobně i poštou vydavatelství MAGNET-PRESS, inzerční oddělení, Vladislavova 26, 113 66 Praha 1, tel. 26 06 51-9, linka 294.

Za původnost a správnost příspěvku odpovídá autor.

Nevyžádané rukopisy nevracíme.

Návštěvy v redakci ve středu od 9 do 16 hodin.

ISSN 0139-7087, číslo indexu 46 044.

Toto číslo má výtah podle plánu 27. 3. 1992.

© Vydavatelství MAGNET-PRESS 1991

Zatím jsme na těchto stranách představili převážně firmy, které byly založeny před druhou světovou válkou a měly tedy po ní na co navazovat, využívat tradice, zkušenosti z minula atd. V tomto směru je firma, kterou vám chceme představit dnes, výjimkou, neboť byla založena roku 1968.

Intel Corporation spatřila světlo světa díky dvěma mužům vědcům, dr. Gordonu Mooreovi a dr. Robertu N. Noyceovi. Jméno společnosti, INTEL, mělo od počátku programový význam pro novou firmu: vytvořeno ze slov INTEGRATED ELectrronics je INTEL téměř symbolem pro jednu z nejrevolučnějších technologií tohoto století – integraci velkého množství diskrétních elektronických součástek na jednom polovodičovém čipu.

Veškeré pozdější úspěchy firmy se opírají o vědeckou práci a zásadní objevy prvního muže firmy INTEL, Gordona E. Moorea, v oblasti principů integrovaných obvodů. Spolu-vynálezcem integrovaných obvodů byl „druhý“ muž firmy, Robert N. Noyce. Ještě v roce založení firmy vstoupil do firmy dnešní prezident (a chief executive officer) dr. Andrew S. Grove, který přispěl značnou měrou při vývoji technologie MOS a jejího využití při výrobě velkých řadí součástek.

INTEL – to bylo kdysi 12 zaměstnanců + management, kteří v prvním roce působnosti dosáhli obratu 2672 amerických dolarů. V obchodním roce 1990 měla firma (koncem roku) 23 900 zaměstnanců po celém světě (z toho asi 1000 v Evropě), kteří uskutečnili obrat 3,9 miliard amerických dolarů (to zhruba odpovídá necelým sedmi miliardám DM), přitom obrat se v uvedeném roce zvětšil oproti roku 1989 o 26 %!

INTEL dnes patří mezi nejprůjnější výrobce v polovodičové „branži“, celosvětově stojí na prvním místě ve výši obratu za mikrosoučástky, vyráběné technologiemi MOS. K této pozici firmy dopomohla především velká produktivita výroby, podpořená nasazováním vždy těch nejnovějších technologií ve všech oblastech. Úspěch pomocí inovací – tak by mohl znít podtitulek tohoto článku, který by plně vystihoval základní koncepci firmy INTEL. O tom, že firma je vedená skutečně na úrovni, svědčí i fakt, že např. v roce 1990 mohla vykázat obrat ve výši 300 000 DM na jednoho zaměstnance.

Již těchto několik čísel a základních faktů ukazuje na jedno: mnohé se ve firmě v době od jejího založení jistě změnilo, avšak původní cíl zakladatelů firmy – využít všech možností mnohaslibné elektronické technologie – v době založení firmy zcela nové – je dodržován až do dnešní doby.

K úspěchu firmy vedla zásadní, nekompromisní a dlouhodobá orientace filosofie podniku: soustředit se na oblast vývoje a výzkumu. Žádý podnik elektronického průmyslu nevkládá tak vysoké procento ročního obratu do výzkumu a vývoje, jako INTEL: v roce 1990 to bylo např. 827 miliónů DM, to je zhruba asi 13 % ročního obratu. Potvrzuje to i investice ve výši 4,2 miliardy DM do nových kapacit v 80. letech, aby byl podnik schopen vyrábět součástky na špičkové technologické úrovni. Podniková politika je prostá – zajistit budoucnost firmy vysokými

investicemi – i proto bylo např. v roce 1991 investováno 1,6 miliardy DM do výstavby nových závodů v New Mexico a Oregonu, jakož i do vybavení moderního výrobního závodu v Irsku. Tato politika je zřejmě správná, protože jak ukazuje skutečnost, jen těmito investicemi byl a je INTEL schopen zajistit si svoji pozici na špičce polovodičového průmyslu a špičkovými výrobky ji dál posilovat.

Historie firmy INTEL je současně také historií neustálých inovací v oblasti polovodičové a počítačové technologie:

1970 . . . první čip LSI – DRAM, 1103 (dynamic random access memory)

1971 . . . první mikroprocesor, 4004, vyroben technologií LSI (large scale integration)

1976 . . . první kompletní jednodeskový počítač iSBC 80/10

1979 . . . mikroprocesor 8088-CPU, umožňující rozvoj osobních počítačů (PC)

od roku 1979 začíná vývoj výkonnějších mikroprocesorů technologií VLSI (very large scale integration) – 8080, 8085, 8086, 8088, 80186, 80188 a 80286, 80386 atd.

1981 . . . první osobní počítač s CPU INTEL

1985 . . . INTEL představuje první 32bitový mikroprocesor 80386. Technologie CMOS umožnila integrovat 275 000 tranzistorů na jednom čipu

1986 . . . uvedení první paměti EPROM 1 MB na trh

1989 . . . i860 CPU – první čip s 1 000 000 tranzistorů

1989 . . . i960 – první čip „superscalar“

1989 . . . i486 – procesor s integrovaným numerickým koprocesorem (1 200 000 tranzistorů na čipu). 50x rychlejší než CPU 8088

1990 . . . i750 – „videočip“ (DVI)

1991 . . . i860XP CPU – čip s 2,55 mil. tranzistorů

Dlouholetý úspěch firmy nelze vysvětlovat pouze technickými přednostmi výrobků, spoluřazujícími pro plnulý vstup jsou jistě i takoví činitelé, jako přizpůsobení se vývoji na trhu, výrobní politika zaměřená na předvídání požadavků trhu, dlouhodobá marketingová strategie a odpovídající vnitřní management. Kromě toho INTEL včas poznal, že není možné se, při stále kratších inovačních cyklech, zabývat pouze vývojem a výrobou polovodičových součástek, neboť uživatelé a výrobci požadují stále lépe připravené funkční skupiny a odbornou podporu. Proto se firma zaměřila na

důsledné sledování potřeb uživatelů.

INTEL vychází vstříc potřebám uživatelů mimo jiné i počítačovými systémy, které zákazníkovi ušetří finančně obvykle velmi náročný vývoj. Skládají se z počítačových desek, provozních systémů a dodatkových softwarových komponent. Uživateli pak stačí připojit jeho specifické hardwarové a softwarové prvky a získá tak výkonná a individuálně přizpůsobená řešení svých potřeb.

Na podporu uživatelů dává INTEL k dispozici ke každému procesorovému výrobku potřebné vývojové prostředky, které umožní, že výrobek může být rychle a optimálně nasazen. To jsou právě ty „vymoženosti“, které vedou k tomu, že jsou výrobky INTEL hojně používány a využívány.

Procesory INTEL a jednočipové počítače mají mnohem širší použití než jen v osobních počítačích. Lze je dnes najít nejen v registračních pokladnách, v ověřovacích a měřicích systémech, v přístrojích pro domácnost, v komplexních řídicích systémech v průmyslu, v moderních kancelářských systémech, v elektronice automobilů, ale i v letadlech (Airbus) a jinde.

Velká změna

Revoluční změny, kterými prochází počítačový průmysl, jsou charakterizovány odvracením se od vlastních řešení k řešením standardizovaným. Uživatelé, kteří chtějí mít své investice jisté, obracejí se k standardním otevřeným systémům. To se týká především architektury čipů, počítačů, provozních systémů a aplikativního software. V současné době se stále méně uživatelů rozhoduje pro vlastní systém. Žádány jsou otevřené systémy, které splňují požadavky na rozšířené standardy a normy, a které umožňují používat výrobky více než jednoho výrobce – univerzální výrobky. Tyto otevřené systémy dají uživateli jistotu účasti na technologickém pokroku a možnost integrovat do svého systému nejnovější a nejvýkonnější výrobky různých specializovaných výrobců.

Toto vše se kryje s iniciativou firmy INTEL zjistit kompatibilitu, plnou služitelnost nových výrobků s dřívějšími a definovat nové standardy. Proto je také např. jeden z nejnovějších procesorů, i486, kompatibilní s procesorem 8088. INTEL navíc úzce spolupracuje a vyměňuje si zkušenosti s pracovníky, kteří vyvíjejí software a je aktivní v různých komisích pro normy.

Již koncem sedmdesátých let poznaly první podniky, že tradiční představy o výpočetní technice mají pouze ohrazenou platnost, a že rozvoj a plnouhodnotnost průmyslu výpočetní techniky lze zajistit pouze zaváděním „všeobjímajících“ standardů. Důležitým krokem ke sjednocení počítačové architektury byla velkosériová výroba mikroprocesorů.

S vitézným postupem standardizace se změnila vertikální struktura počítačového průmyslu na horizontální. Pro počítače průmyslového standardu jsou na trhu různé provozní systémy, nespočetné uživatelské programy, různé typy periferií zařízení a zařízení, rozšiřujících možností počítačů. Tím, že se používají existující standardy, mohli se výrobci vzdát mnohých vlastních konstrukcí a použít ušetřené investice na vývoj inovativních technologií. INTEL tak přispěl ke změně počítačového průmyslu, ke zvětšení výkonu počítačů a tím i k větší jistotě, že investice uživatelů budou vynaloženy co nejoptimálněji.

INTEL dnes

INTEL v současné době zaměstnává po celém světě kolem 24 000 lidí, z toho něco více než 1000 v Evropě. Ústředí společnosti je v městečku Santa Clara v Kalifornii, USA,

sídlo INTEL GmbH Deutschland je ve Feldkirchenu u Mnichova. Další důležitá obchodní zastoupení a výrobní závody jsou v Hillsboro, Phoenixu, Rio Rancho, Quebechu, Puerto Rico, Paříži, Madridu, Swindonu, Rotterdamu, Curychu, Miláně, Štokholmu, Helsinkách, Jeruzalémě, Penangu, Singapuru, Hongkongu, Manile, Tokiu. INTEL je zastoupen ve více než 20 zemích světa a má více než 80 obchodních zastoupení.

Změna počítačového průmyslu má však nejen technologický charakter. Na začátku 90 let byly výrobní podniky (a jsou dodnes) vystaveny různým dramatickým strukturálním změnám. Asijsko-pacifický trh je charakterizován velkým boomem, společný evropský trh je přede dveřmi a východní Evropa se stále více otevírá Západu. Již v roce 1990 vyrabil INTEL 46 % svého obratu mimo území Severní Ameriky – významný trend, který bude zřejmě s budoucím vývojem nabývat na stále větší důležitosti.

INTEL chápá změnu struktury počítačového průmyslu jako šanci a výzvu. Jako vůdčí firma v nabídce počítačových součástí a systémů umožňuje INTEL uživatelům na celém světě podílet se na obrovském pokroku „počítačového věku“.

Autorizovaným distributorem firmy INTEL v ČSFR je firma Bacher Electronics, Přechodní 11/1600, 140 00 Praha 4 – Krč, telefon (02) 692 80 87, tel., fax (02) 471 80 03.

A na závěr trochu prognostiky. Znáte Moorův zákon? Na jeho základě byla vytvořena předpověď pro složitost integrovaných obvodů do roku 2000 (podle Gordona Moorea, prezidenta INTEL). Vezměme to však popořádku: budeme-li analyzovat technické trendy (technologie), které umožnily konstruovat a vyrábět stále výkonnější integrované obvody, zjistíme, že stále rostoucí počet tranzistorů na jednom čipu v poměru k odpovídající ploše čipu je za posledních 20 let víceméně konstantní číslo. Tento vztah vyjadřuje právě tzv. Moorův zákon, který jeho tvůrce vyjádřil již před mnoha lety: Počet tranzistorů v jednom mikroprocesoru se každě dva roky zdvojnásobí.

Čím je to umožněno? Tady právě vstupují do hry nové technologie, které umožňují jak

zkušovat nutné spoje na čipu, tak zmenšovat plochu elektrod tranzistorů. Samozřejmě, čím jsou užší spoje a menší plochy elektrod, tím je jednak vyšší možný pracovní kmitočet a jednak se na danou plochu vejde větší počet tranzistorů. Populárně lze uvést, že byla-li tloušťka spojů na čipu před asi 20 lety zhruba rovna tloušťce vlasu, odpovídá dnes délce průměrné bakterie, tj. asi 1 μm (milionina metru). Přitom se předpokládá, že do konce století se dále zmenší asi na 1/5 μm, při vývojových pracech v laboratořích lze předpokládat, že bude dosaženo tloušťky asi 1/10 μm.

Stále se zmenšující rozměry tranzistorů a spojů tedy umožňují zvětšit jejich počet na dané ploše čipu, navíc mohou být zpracovávány signály vyšších kmitočtů – to působí příznivě na rychlosť, s niž mikroprocesor může zpracovávat data a instrukce. Proto moderní procesory pracují dnes obvykle s taktem (hodinovým kmitočtem) mezi 25 až 40 MHz.

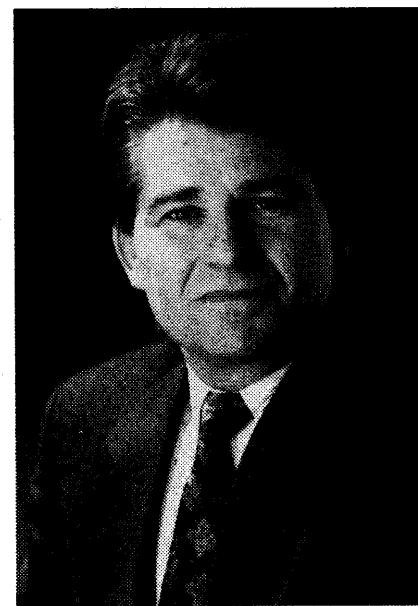
Je zřejmé, že cíl vývojářů – dostat na danou plochu co největší počet tranzistorů – je pouze jedním z ukazatelů jakosti výroby. Druhým je bezesporu spolehlivost. Zlepšit spolehlivost je cílem inženýrů a techniků, připravujících výrobu. I v této oblasti dosáhl INTEL vynikajících výsledků, počet vadných čipů při výrobě se stále zmenšuje (býval zhruba v průměru asi 200 z milionu zhotovených kusů). V současné době se blíží nula – to umožňuje vyrábět čipy stále větší velikosti a složitosti.

Z toho, co jsme uvedli, jasně vyplývá prognóza složitosti mikroprocesoru pro konec století, z této složitosti pak i jeho vlastnosti:

100 milionů tranzistorů na čipu, plocha čipu 6,45 cm², taktovací kmitočet 250 MHz, 2 miliardy instrukcí za sekundu a navíc INTEL zaručuje, že takovýto mikroprocesor budoucnosti bude plně kompatibilní se stávajícím mikroprocesorem i386. Jen pro srovnání: dnes vyráběný mikroprocesor i486 má 1,2 milionu tranzistorů, taktovací kmitočet 33 MHz, 27 milionů instrukcí za sekundu – z tohoto srovnání si lze udělat zhruba představu, o jak výkonný procesor půjde a co všechno umožní.

Stále se zvětšující integrace povede samozřejmě ve svých důsledcích i k cíli všech výrobců počítačů – umožní konstruovat a vyrábět osobní počítač s jedním jediným čipem. Cesta k tomuto cíli je nastoupena – osobní počítač v roce 1984 měl asi 170 jednotlivých čipů plus pracovní paměť. V roce 1987 se počet čipů zmenšíl asi na 70 plus pracovní paměť, když INTEL uvedl na trh v roce 1990 mikroprocesor 386SL, umožnil výrobu osobních počítačů, které měly kolem 10 čipů plus pracovní paměť. V roce 1993 bude tedy zřejmě možné vyrobit PC dnešních vlastností s jedním jediným čipem. Co to pro výrobu osobních počítačů znamená, není třeba rovnádět.

I když je INTEL pouze jednou z firem v širokém spektru výrobců polovodičových součástek, jeho význam pro rozvoj elektroniky a zvláště výpočetní techniky byl a je zcela jedinečný a to po všech stránkách – jak po stránce technické, tak i organizační, obchodní atd. Všechny podrobné informace o firmě INTEL a jejich výrobcích lze získat, jak bylo uvedeno, u firmy Bacher Electronics, která je výhradním distributorem INTEL. L.K.



Joachim Rissmann, obchodní vedoucí INTEL GmbH v Mnichově

MĚŘENÍ ELEKTRICKÝCH VELIČIN NA POČÍTAČI PC

Ing. Josef Petřík

Příspěvek obsahuje podrobné stavební návody na stavbu zařízení, která umožní měřit, zpracovávat a zobrazovat výsledky měření na počítačích řady IBM PC nebo po úpravě programového vybavení i na jiných počítačích, vybavených paralelním rozhraním pro připojení tiskárny (CENTRONICS).

Počítače PC mají možnost připojit 3 tiskárny, označené LPT1, LPT2 a LPT3.

Standardně bývá zapojen a instalován konektor pro připojení tiskárny LPT 2, pro níž je datový registr přístupný na adrese 378 (hexa), stavový na adrese 379 a řídicí na adrese 37A (hexa).

Všechny programy v tomto návodu jsou pro tento stav. V některých případech, např. při použití adaptéru Hercules color grafic card, je změněna adresa tiskárny na LPT1, pro níž je datový registr přístupný na adrese 3BC (hexa). Datový registr LPT3 má adresu 278.

Informace o nastavené adrese se nachází v SETUP počítače (u některých ne).

V případě adresy datového portu 3BC je nutné změnit adresy v programovém vybavení takto:

PORT [\$378] za PORT [\$3BC]
PORT [\$379] za PORT [\$3BD]
PORT [\$37A] za PORT [\$3BE]

V případě adresy datového portu 278 je nutné změnit adresy v programovém vybavení takto:

PORT [\$378] za PORT [\$278]
PORT [\$379] za PORT [\$279]
PORT [\$37A] za PORT [\$27A]

Budete-li chtít použít jiné typy počítačů, vybavené rozhraním CENTRONICS, je nutné znát adresy výše uvedených registrů a provést popsanou úpravu programového vybavení.

Uváděné návody obsahují jak jednoduchá zařízení pro měření napětí, která může zhotovit i nepříliš zdatný amatér za cenu asi 100 Kčs během jedné hodiny, tak zařízení obsahující dvanáctirozahový měřicí přístroj s automatickým přepínáním rozsahů, automatickou kompenzací změn elektrických parametrů obvodových prvků, s automatickou indikací polarity měřených veličin a automatickou indikací překročení měřitelných parametrů – vše za cenu, která nepřekročí asi 300 Kčs bez nutnosti shánět speciální součástky.

Uvedené programové vybavení umožní uživateli další tvůrčí práci s popisovaným zařízením využitím určitých rutin a procedur.

V příspěvku uvedená obvodová schémata jsou navržena s minimem součástek a s ob-

vodovými prvky takřka nejnižších cenových hladin; horší vlastnosti součástek jsou kompenzovány možnostmi počítače při zpracování výsledků měření. Minimum součástek vede také k využívání vlastních přirozených vlastností těchto obvodových prvků.

Úvod

Výpočetní technika je dosud užívána převážně pro zpracování programů řešících matematické, logické nebo textové či grafické problémy, případně graficko-početní operace. Možnosti i osobních počítačů jsou však mnohem větší. Dovolují ve spojení s vhodnými periferními zařízeními zpracovávat údaje o stavech nebo změnách stavů např. fyzikálních veličin, případně po jejich zpracování v počítači působit přes příslušná periferní zařízení zpětně na sledované jevy.

Pro účely elektronického zpracování je nutné všechny tyto fyzikální veličiny převést na pokud možno jednotný elektrický signál v převodníkách fyzikálních veličin. Tento analogový elektrický signál je nutně dále pro zpracování počítačem převést do číslicové formy v analogově-číslicových převodníkách, nebo naopak převést diskrétní číslicový signál z počítače na analogový číslicově-analogových převodníkách.

Realizace převodníků

Všechny převodníky uvedené v tomto návodu vyžadují stabilizované napájecí napětí ± 15 V. Z důvodu bezpečnosti byl jako zdroj uvažován školní zdroj BK 125, který by měl být na školách a v odborných kroužcích běžně dostupný. (V současné době je inzerován výrobcem, TESLA Brno, za prodejnou cenu 490 Kčs – viz AR-A č. 1/1992.) Je možné ovšem použít libovolný zdroj napětí ± 15 V, při realizaci pouze převodníku A/D může být vzhledem k malému proudovému odběru vestavěn zdroj do síťové vidlice („kalkulačkový“ typ).

Jednoduché převodníky D/A a A/D lze realizovat na nepájivých kontaktních polích, prodávaných za asi 30 Kčs např. v prodejnách bývalého DOSS Svažarm (i na dobírku), popř. u firmy Diametral, Vinohradská 170, 130 00 Praha 3.

Složitější převodníky jsou postaveny na „univerzální“ desce s plošnými spoji podle obr. 12.

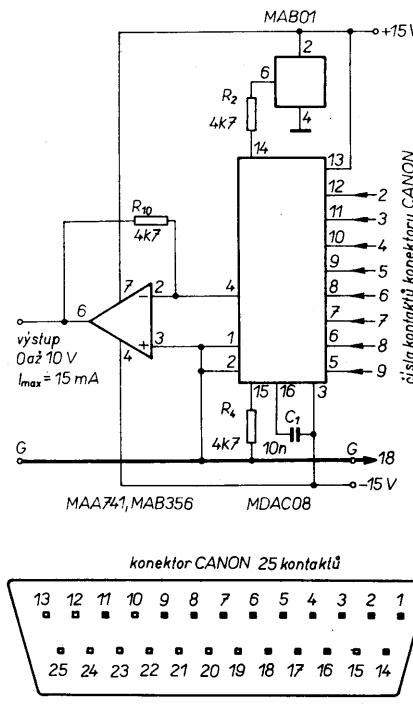
25polový konektor CANON, určený pro připojení tiskárny k počítačům PC, lze zakoupit v prodejnách s elektronickými součástkami nebo příslušenstvím počítačů za asi 39 Kčs.

Tamtéž lze koupit plochý, minimálně 14žilový vodič buď naši výroby (nejenči) délky maximálně 1 m za asi 11 Kčs, nebo originální s roztečí 1,25 mm za asi 30 Kčs, který je nutný při použití „primořezného“ konektoru pro výstup z desky s plošnými spoji u nejsložitějšího zapojení.

Nejjednodušší převodník D/A na nepájivém poli

(zapojení bez zdroje referenčního napětí nebylo uvažováno)

Na obr. 1 je obvodové schéma převodníku D/A, který umožňuje programově nastavit na



Obr. 1. Schéma nejjednoduššího převodníku D/A

Obr. 2. DA1 – program řízení převodníku z obr. 1 Kurzorovými klávesami

```

var
  x : byte;
  ch : char;

procedure Hlaseni(St:string);
begin
  gotoXY(1,20);
  ClrScr;           {smaze radek od cursoru do konca}
  writeln(St);
end;

begin
  ClrScr;
  writeln('Rízení D/A převodníku pomocí kurzorových kláves:');
  writeln(' 1'.chr(24), '1' - zvyšování napětí');
  writeln(' 1'.chr(25), '1' - snižování napětí');
  writeln('Napětí se mění od 0 do +10 V po skocích cca 40 mV');
  writeln;
  writeln('Zadávání lze přerušit stiskem klávesy [Esc]');
  PORT1[$378]:=2;
  x := 0;
  repeat
    Ch := ReadKey();
    if Ch = #27 then HALT; {stisknut Escape}
    if Ch = #0 then
      begin
        Ch := ReadKey();
        if Ch = #72 then {sípka nahoru}
          begin
            if x < 255
              then
                begin
                  inc(x);Hlaseni('');
                end
              else
                Hlaseni('U = Umax');
            end;
        end;
        if Ch = #80 then {sípka dolu}
          begin
            if x > 0
              then
                begin
                  dec(x);Hlaseni('');
                end
              else
                Hlaseni('U = Umin');
            end;
        end;
        GotoXY(1,10);writeln('Krok = ',x:3);Port1[$378]:=x;
        GotoXY(1,12);writeln('Napětí je asi = ',10/255 * x:3,' V');
      end;
    until false;
  end.

```

výstupu napětí 0 až +10 V v 255 krocích po asi 39 mV.

Převodník D/A obsahuje zdroj referenčního napětí 10 V (MAB01), který spolu s R_2 a R_4 vytváří referenční proud asi 2 mA pro vlastní převodník MDAC08. Vstupní data pro převodník v binární soustavě se přivádějí z datového registru pro řízení tiskárny (adresa 378H pro standardní provedení).

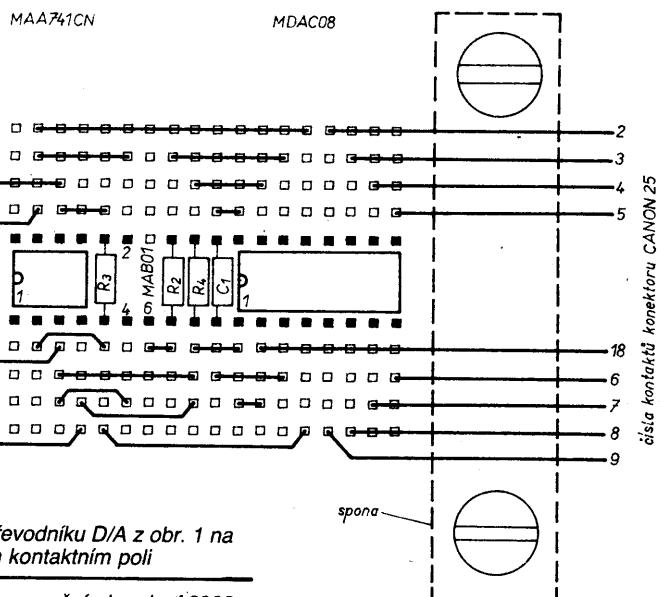
Poněvadž výstup MDAC08 je proudový, tj. binárnímu číslu na vstupech 5 až 12 odpovídá proud mezi vývody 4 a 2, následuje převodník proud-napětí z operačního zesilovače MAA741 nebo MAB356 apod.

Komunikace počítače s převodníkem D/A je velice jednoduchá, v jazyce GWBASIC příkazem OUT & H378, 255 bude na výstupu napětí 10 V, příkazem OUT & H378, ... (jiné číslo v rozsahu 0 až 255) bude na výstupu odpovídající napětí.

V jazyce TURBOPASCAL je komunikace zabezpečena příkazem PORT [\$378]:= číslo 0 až 255. Na obr. 2 je uveden program DA1, který nastavuje napětí převodníku D/A pomocí kurzorových kláves

↑ ↓.

Převodník je zapojen na nepájivém kontaktním poli podle obr. 3. Tlusté čáry jsou propojovací vodiče, součástky jsou přímo

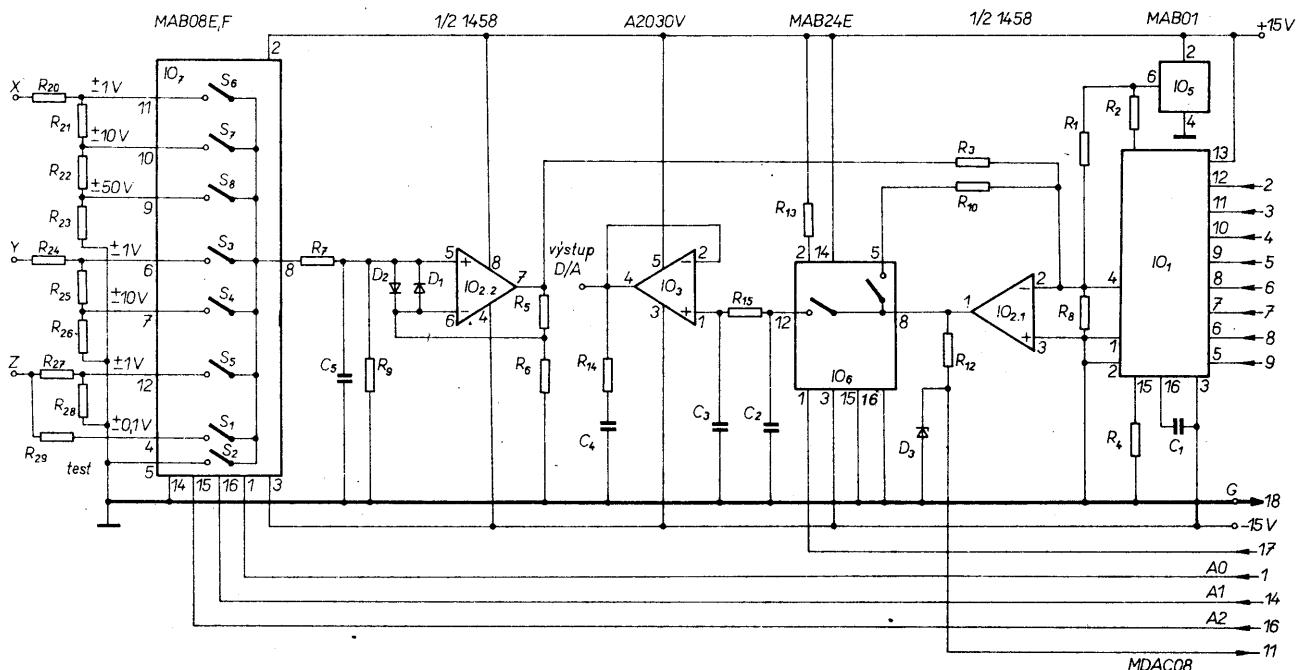


Obr. 3. Zapojení převodníku D/A z obr. 1 na nepájivém kontaktním poli

Obr. 4. Fotografie upevnění obvodu A2030 (4. str. obálky)

zasunuty do děr v kontaktním poli. U obvodu MAB01 jsou nevyužité vývody odštípnuty. Operační zesilovač MAA741 použijeme poukud možno typu CN (v pouzdro dual-in-line).

Pokud pro propojení s konektorem CANON použijeme plochý vodič s lankami, pocinujeme jeho konec, které se zasunou do kontaktního pole. Zvýšenou pozornost je třeba věnovat



Obr. 5. Schéma tříkanálového převodníku A/D a D/A bez MAB311

pájení na konektoru CANON (25 vývodů), aby nevznikly zkraty mezi vývody nebo aby nebyly vývody přehozeny. Je třeba si uvědomit, že stačí jediný chybný spoj nebo kontakt a zařízení nefunguje. (Na rozdíl od zahrádkářů, kteří zasejí-li mrkev a vyrosté květák, tak přece jen něco vyrostlo.) Pokud si příliš nedůvěřujeme, je vhodné navléci přes pájné přívody izolační bužírky (obr. 4).

Na kontaktním poli je vhodné plochý vodič upevnit sponou z kousku plechu, aby se při manipulaci zabránilo vytření vodičů z dřív kontaktním poli (viz obr. 3).

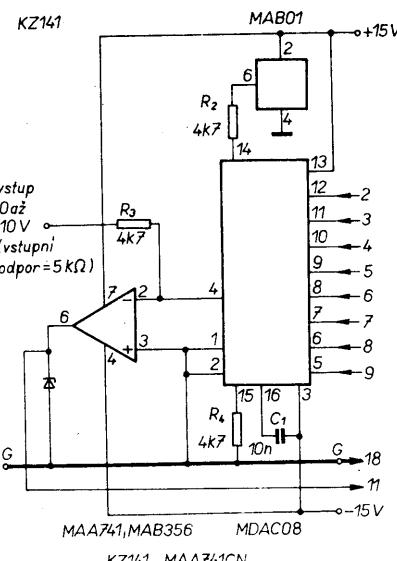
Zhotovené zařízení můžeme upevnit do krabičky s výstupními svorkami, nebo může být sestavováno jako laboratorní úloha. Nevhodou uvedeného zapojení je malý prourový odběr z tohoto programovatelného zdroje napětí, asi 15 mA. Zvětšit výstupní proud umožní výkonový obvod A2030 v zapojení podle obr. 5 nebo zkušenější je mohou zapojit přímo jako převodník proud – napětí (místo MAA741).

Náměty pro využití zhotoveného přístroje jsou uvedeny v poslední části článku.

Nejjednodušší převodník A/D

Schéma převodníku A/D je na obr. 6. Převodník obsahuje opět zdroj referenčního proudu MAB01 s rezistory R_2 a R_4 , převodník D/A MDAC08 a komparátor, tvořený operačním zesilovačem MAA741 nebo MAB356 s otevřenou smyčkou zpětné vazby.

Komparátor porovnává proud generovaný převodníkem D/A a proud vytvářený vstupním napětím přes rezistor R_3 . Podle výsledku komparace se mění úroveň digitálního signálu (na výstupu komparátoru), který se vede do počítače přes kontakt 11 konektoru CANON (signál BUSY).

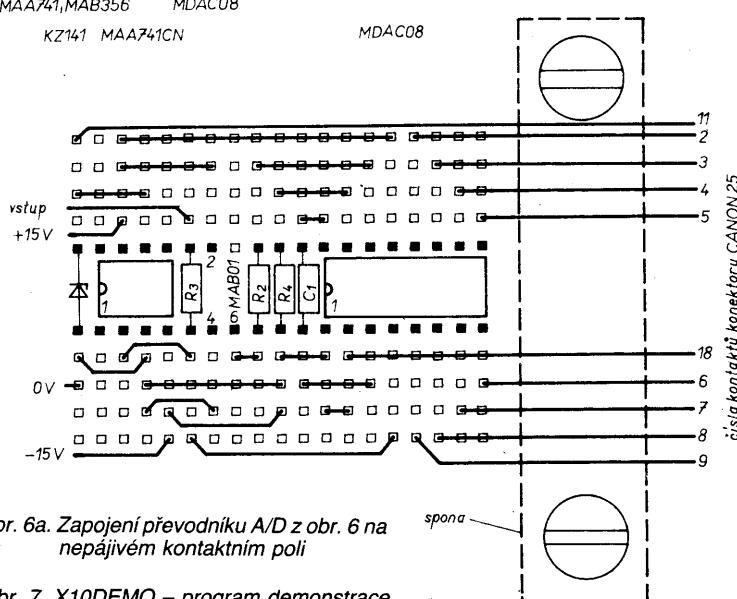


Obr. 6. Schéma nejjednoduššího převodníku A/D

Počítač komunikuje s převodníkem v jazyce GWBASIC pomocí programu AD2B a AD-2BTTEST, v jazyce TURBOPASCAL AD2P a AD2PTTEST.

Vlastní algoritmus approximace je uložen v proceduře APROX, při níž počítač generuje slovo do převodníku D/A podle výsledku komparace. Blíže o approximaci v lit. [5] – protože podle známého pravidla je jeden obrázek lepší než tisíc slov, je na obr. 7 program pro demonstraci funkce postupné approximace a na obr. 8 je kopie obrazovky z tohoto programu.

Nevýhodou tohoto převodníku je malý vstupní odpor asi $5\text{ k}\Omega$, zatěžující zdroj měřeného napětí.



Obr. 6a. Zapojení převodníku A/D z obr. 6 na nepájivém kontaktním poli

Obr. 7. X10DEMO – program demonstrace algoritmu postupné approximace

```

program X10DEMO; (KANAL X ROZSAH 10 V)
uses CRT,GRAPH;

const
  RefNapetiX10 = 9.78;
  BigFont = SmallFont; {globální nastavení fontu}

var
  Ch : char;
  K,A,i,d,Ap,Vstup : byte;
  BX : real;
  p, Gd, Gm : integer;
  St : string[80];

procedure bar;
begin
  setcolor(15);
  SetFillStyle(1,15);
  Bar3D(600-70*i,300,540-70*i,300-A,4,true);
end;

procedure velikostA;
begin
  St := ''; str(A:3, St);
  SetColor(1); SetTextJustify(CenterText, CenterText);
  SetTextStyle(DefaultFont, HorizDir, 2);
  OutTextXY(570-70*i, 290, St);
end;

procedure DemoAprox;
begin
  A := 0; Vstup := 0;
  for i := 7 downto 0 do
  begin
    Port[$3781] := A + 1 shl i; delay(400); Vstup := Port[$3781];
    if Vstup > 128 then
    begin
      delay(1); A := A + 1 shl i; bar; velikostA;
    end;
  end;
  bar; velikostA;
end;

procedure Aprox;
begin
  A := 0; Vstup := 0;
  for i := 7 downto 0 do
  begin
    Port[$3781] := A + 1 shl i; delay(1); Vstup := Port[$3781];
    if Vstup > 128 then A := A + 1 shl i;
  end;
end;

procedure X;
begin
  Port[$37A1]:=10; Aprox; K:=122;
  if K=255 then K:=254; Port[$37A1]:=13; DemoAprox;
  BX:=round(RefNapetiX10 *100*(A-K)/(255-K))/100;
end;

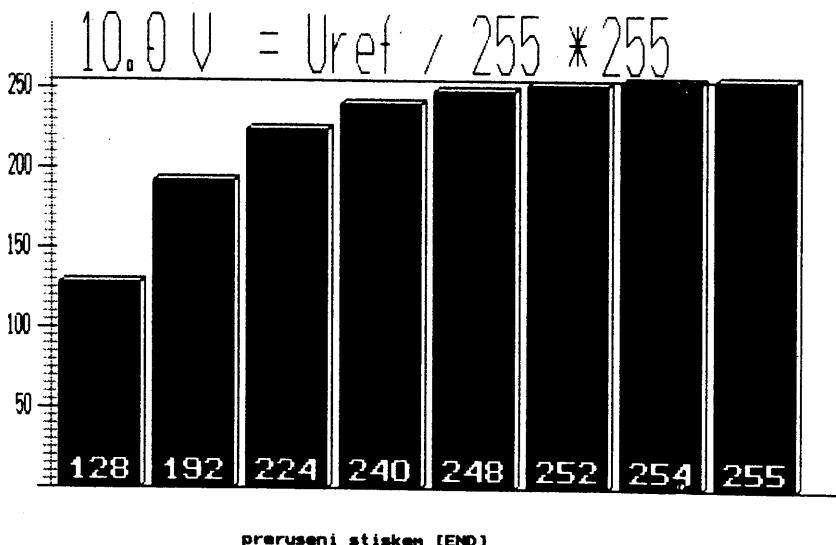
```

```

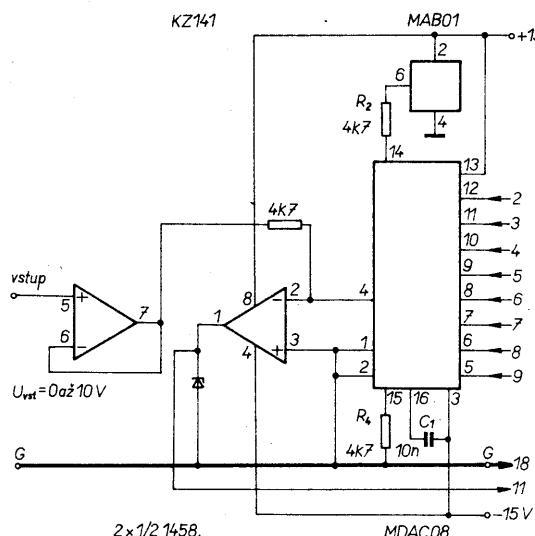
procedure ZobrazNapetix;
var
  St: String[40];
BEGIN
  SetTextStyle(BigFont, HorizDir, 10); SetUserCharSize(3,1,6,1);
  SetColor(14); SetTextJustify(lefttext, bottomtext); {barva údaje}
  str(50:1,St); OutTextXY(50,40,St); OutTextXY(150,40,'V');
  OutTextXY(200,40,'= Uref / 255 * '); str(5, St); OutTextXY(420,40, St);
end;

{počátek programu}
BEGIN
Gd := Detect; InitGraph(Gd, Gm, '');
if GraphResult <> grOk then
begin
  writeln('Nelze prepnout do grafiky.');
  Halt(1);
end;
ClearDevice; SetColor(2);
for i:=9 to 60 do
Line(38,5*i,48,5*i); SetColor(0); SetTextStyle(BigFont, HorizDir, 10);
SetUserCharSize(1,1,2,1);
repeat;
if KeyPressed then
begin
  Ch:=ReadKey;
  if Ch#0 then
  begin
    Ch:=ReadKey;
    if Ch#79 then Halt;
  end;
end;
SetColor(15); SetTextStyle(DefaultFont, HorizDir, 1);
OutTextXY(100,337,'prerusení tiskem [END]');
SetTextStyle(BigFont, HorizDir, 1); SetUserCharSize(1,1,2,1);
SetColor(2); Line(44,10,44,300);
SetColor(2); Line(38,5*i,48,5*i); SetColor(15);
for i:=9 to 60 do
Line(38,5*i,48,5*i); SetColor(15);
for i:=1 to 5 do
begin
Line(34,58*i,48,58*i); Line(34,300,648,300);
SetUserCharSize(1,1,2,1); SetTextJustify(lefttext, centerText);
str(50*i:5,St);
OutTextXY(8,297-58*i,St);
end;
x:ZobrazNapetix; Line(20,300-a,600,300-a); delay(2000); ClearDevice;
until keyPressed;
CloseGraph;
END.

```



Obr. 8. Kopie obrazovky z programu X10DEMO – postupná approximace



Obr. 9. Schéma převodníku A/D s MA1458 a zesílením 1

Tuto nevýhodu odstraňuje zapojení na obr. 9, obsahující navíc napěťový sledovač z jedné poloviny obvodu MA1458 (nebo B082, MAB412 apod.).

Realizace tohoto převodníku na nepájivém poli je na obr. 10 a programové vybavení je stejně jako pro obvod na obr. 6. Napětí ve voltech, získané převodem, je v programu AD2P zobrazeno pouze v textovém režimu. Zobrazení velkými číslicemi v grafickém režimu může uživatel provést sám, např. podle programu XYZRG.

Oba uvedené převodníky vyžadují nastavení, které spočívá ve stanovení programové konstanty RefNap, určující velikost vstupního napětí, odpovídajícímu číslu získanému převodem 255. Ke stanovení RefNap použijeme program AD2PTEST nebo AD-2BTTEST a schéma pro nastavení převodníků na obr. 11.

Schéma obsahuje regulovatelný zdroj napětí 0 až minimálně +10 V (např. laboratorní

zdroj BK 127 nebo zdroj sestavený ze zdroje pevného napětí +15 V a jednoho – nebo pro přesnejší nastavení – dvou potenciometrů asi 1 k Ω (logaritmických).

Napětí v rozsahu 0 až +10 V přivádíme na vstup převodníku a sledujeme na obrazovce velikost čísla, získaného převodem. Při dosažení 255 přečteme na voltmetru odpovídající napětí, které zadáme jako konstantu RefNap do programu AD2P nebo AD2B. Dále využíváme jen těchto programů.

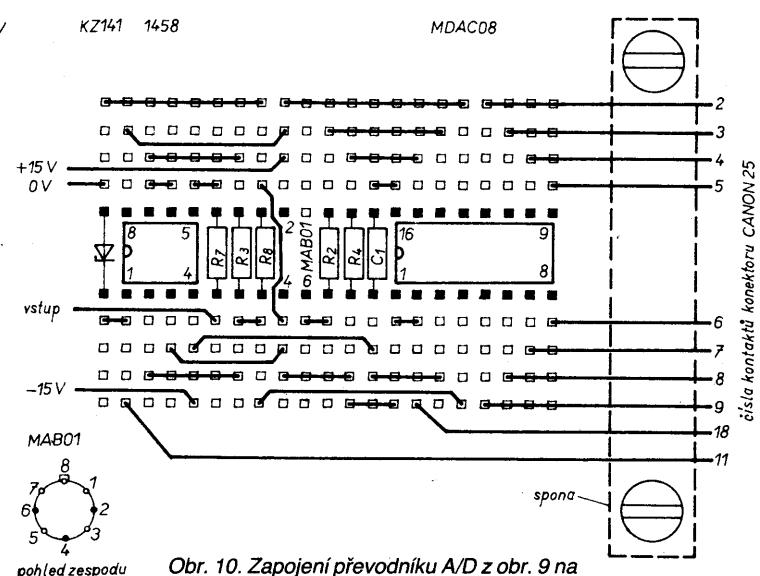
Pro tvorbu vlastního uživatelského programu používáme programovou rutinu APPROX, která odevzdává proměnnou A úměrnou velikosti vstupního napětí v rozsahu 0 až 255. Získání již zaokrouhlené hodnoty napětí je vidět na dalším rádku programu. Blíže o tvorbě vlastního programového vybavení v kapitole Měření a vlastní uživatelské využití.

Převodníky A/D a D/A na „univerzální“ desce s plošnými spoji

Deska s plošnými spoji na obr. 12 umožňuje konstruovat několik typů zařízení. Nejprve bude popsáno nejsložitější provedení a potom odvozené jednodušší varianty, které vzniknou pouhým vynecháním některých součástek nebo jejich záměnou.

3kanálový osmibitový převodník A/D s automatickým přepínáním rozsahu, automatickou kompenzací posuvu napěťové nesymetrie a posuvu bipolární nuly, automatickou indikací polarity měřeného napětí, automatickou indikací překročení měřitelných parametrů a výkonový převodník D/A ve funkci programovatelného zdroje napětí do ± 10 V

Schéma zařízení je na obr. 13 a představuje pravděpodobně nejlevnější variantu ú-

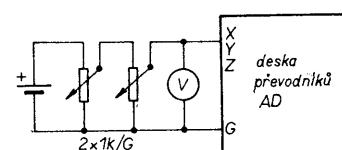


Obr. 10. Zapojení převodníku A/D z obr. 9 na nepájivém kontaktním poli

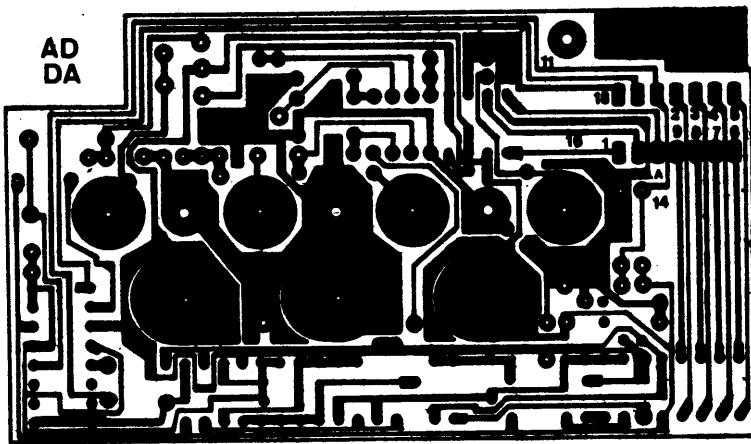
plného periferního zařízení vhodného pro školní a amatérské využití počítačů PC v měřicí technice.

Zařízení bylo opět navrženo a odzkoušeno s minimem součástek (navíc běžných na našem trhu) tak, aby jej mohli realizovat i zájemci nepříliš zdatní v elektronice.

Obsahuje osmibitový převodník D/A, MDAC08, řízený datovým portem tiskárny. Převodník je použit jednak pro nastavení výstupního napětí programovatelného zdroje, jednak v převodníku A/D ke generování proudu pro komparátor při postupné approximaci. Dále obsahuje zdroj referenčního proudu tvořený obvodem IO₅ a rezistory R₂, R₄, komparátor IO₄, převodník proud-napětí IO_{2,1}, impedanční převodník IO₃ a vstupní zesilovač IO_{2,2} se zesílením přibližně 100. Komparátor MAB311 byl použit z důvodu rychlosti převodu A/D.



Obr. 11. Schéma pro nastavení převodníků A/D

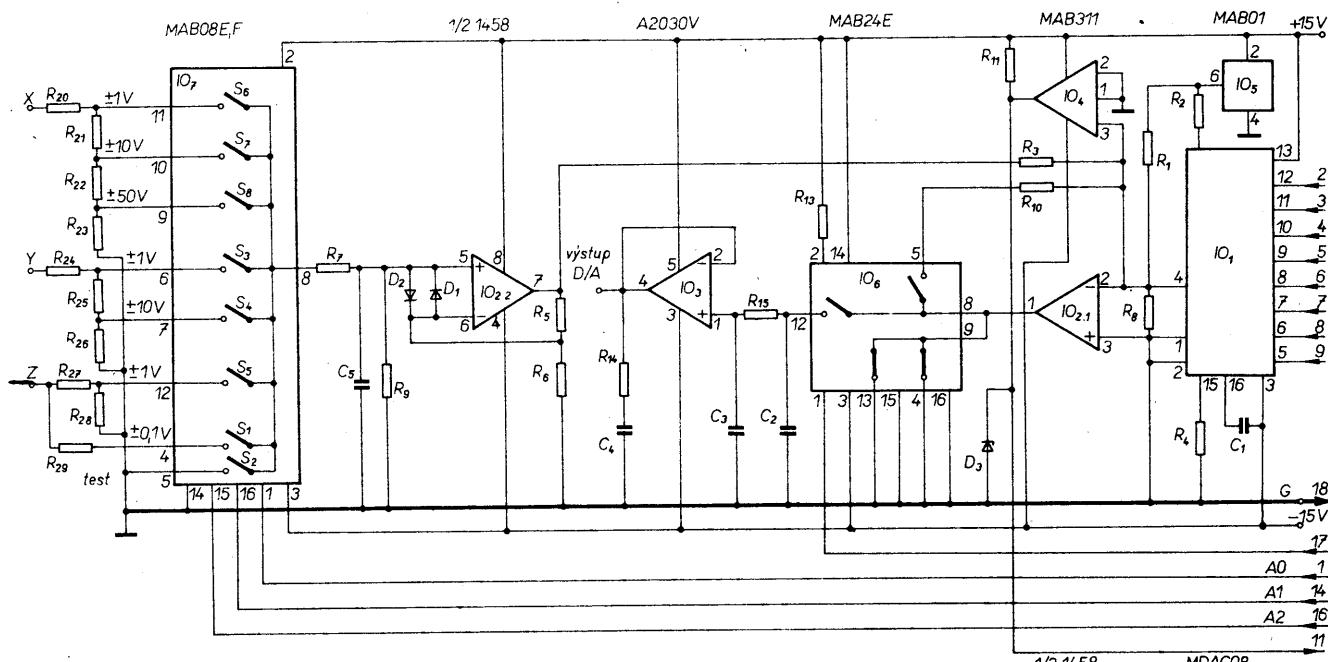


Obr. 12. „Univerzální“ deska s plošnými spoji (A200)

Před každým měřicím cyklem je sepnut kontakt S_2 vstupního multiplexeru, který spojí vstup $IO_{2,1}$ s nulovým potenciálem a pak proběhne převod A/D. Číslo získané převodem se pak odečítá od hodnoty získané při měření ve zvoleném kanálu. Zároveň počítač zaokrouhluje hodnotu získanou měřením na reálný počet míst.

Přesnost měření je $\pm 1\%$ ze zvoleného rozsahu a doba převodu pro počítač AT s kmitočtem hodin 12 MHz v jazyce TURBO-PASCAL je asi 150 mikrosekund. (Pro srovnání doba převodu podobných jednotek typu NTL řady CATT nebo typu NEVA – typ 7820 – v cenách od 375 do 1000 DM je 100 mikrosekund ve strojovém kódu).

V režimu „převodník A/D“ se povelem pro multiplexer IO_6 uzavře smyčka zpětné vazby IO_1 , který pak funguje jako převodník proudnapětí. Rozpojí se zkratovací kontakty obvodu IO_6 a napětí na výstupu $IO_{2,1}$, úměrné zadanému slovu na vstupu obvodu MDAC08, se vede na analogovou paměť.



Obr. 13. Schéma tříkanálového převodníku A/D a D/A s MAB311

Multiplexer IO_7 je řízen 3 bity řídicího registru tiskárny a slouží pro přepínání vstupních kanálů a rozsahů.

Multiplexer IO_6 je řízen dalším bitem řídicího registru a přepíná režim A/D a D/A.

Na obr. 13 je zobrazen stav v režimu A/D a měření v kanálu X na rozsahu ± 1 V. Obvod $IO_{2,1}$ má rozpojenou smyčku zpětné vazby a je dalším kontaktem multiplexeru odpojen od analogové paměti převodníku D/A, tvořené rezistorem R_{15} a kondenzátorem C_3 . Výstup obvodu $IO_{2,1}$ je připojen na nulový potenciál dvěma paralelně spojenými kontakty multiplexeru (pro zmenšení hodnoty R_{on}), aby svým výstupem neovlivňoval vlivem konečné velikosti odporu multiplexeru v rozepnutém stavu vstup komparátoru. Kondenzátor C_2 zmenšuje zvlnění na výstupu převodníku D/A pro delší dobu převodu A/D.

Vstupní napětí z kanálu X je vedené přes rezistor R_{20} (100 k Ω), který tvoří část vstupního děliče a současně působí jako ochrana spolu s diodami D_1 , D_2 proti napěťovému

přetížení vstupních obvodů do 100 V (viz lit. [5]). Podobně fungují i rezistory R_{24} , R_{27} a R_{29} v dalších kanálech.

Tyto rezistory tvoří spolu s kondenzátorem C_5 filtr pro rychlé napěťové špičky ve vstupním napětí. Kapacitu filtracního kondenzátoru nezvětšujeme, protože časová konstanta musí být malá vzhledem k programově zajištěné rutině testu na změnu elektrických parametrů použitých obvodů. Odpor tétoho rezistoru určuje vstupní odpory jednotlivých kanálů; byly zvoleny s ohledem na praktické použití a i s ohledem na zbytky měření vlivem svodových proudů multiplexeru. Blíže o návrhu obvodových prvků z hlediska přesnosti měření viz lit. [7].

Zesílený signál v $IO_{2,1}$ je veden přes rezistor R_3 na komparátor IO_4 .

Napěťová nesymetrie $IO_{2,1}$ i její posuv a základní nastavení bipolární nuly i její změna u převodníku D/A MDAC08 je kompenzována programově, takže se jednak nemusí použít prvky pro nastavování a jednak jsou programově kompenzovány i teplotní a dlouhodobé změny elektrických parametrů.

R_{15} , C_3 a dále přes impedanční převodník IO_3 na výstup převodníku D/A. Zároveň sepnut kontakt S_2 multiplexeru IO_7 , který uzemní vstup $IO_{2,1}$, takže na jeho výstupu je přiblžně 0 V a není ovlivňován převod D/A.

Casová konstanta analogové paměti je volena tak, aby zvlnění výstupního napětí převodníku D/A bylo minimální i pro další provedení s delší dobou převodu (bez komparátoru MAB311).

Při uživatelské vlastní programové využití musí být pak doba nastavení výstupního napětí převodníku D/A delší než součin $R_{15} \cdot C_3$ a je realizovatelná příkazem **DELAY**.

Analogová paměť je využívána pouze po dobu vlastního převodu A/D, každá programová rutina APROX končí povelem pro návrat do režimu D/A.

Uvedené zapojení s analogovou pamětí bylo zvoleno z důvodu jednoduchosti, minimální součástek a napájecího napětí jen 15 V.

Byla uvažována i dokonalejší řešení s modernějšími obvody – vzhledem k dostupnosti a ceně bylo zvoleno toto řešení.

Konstrukční provedení

Všechny součástky převodníků včetně vstupních a výstupních svorek jsou na jedné desce s plošnými spoji, která dovoluje značnou variabilitu konstrukčního uspořádání. Osazení desky je poměrně náročné na čistotu pájení i na mechanickou výrobu. Pěče věnovaná těmto operacím se vrátí v bezchybné funkci zařízení. Pokud si nejsme jisti svoji prací, osadíme nejprve desku bez obvodu A2030 a pomocí programu na obr. 14 oživíme část A/D. Pak doplníme A2030.

Za základní byla zvolena konstrukce s běžně dostupným „úředně“ bezpečným napájecím zdrojem BK 125, který má kromě požadovaného napájecího napětí ± 15 V ještě i napětí $+5$ V.

Zvolená konstrukce (na obr. 18) umožňuje využívat zdroje i při provozu jednotky převodníků a bude využita v dalších rozšiřujících přímozávazných modulech, např. v jednoduchém programovatelném generátoru harmonických napětí.

V tomto konstrukčním provedení jsou do desky s plošnými spoji vyvrty díry o průměru 11,5 mm a celá deska je nasazena na stávající přístrojové zdírky zdroje BK 125, posunuta směrem dolů a dotážena otáčivou částí svorky. Vytvořené spojení je pevné a spolehlivé, zvláště budou-li spoje na desce pocínovány (dosedací plošky popř. pocinujeme opatrně sami. Pozor – po cínování je třeba odstranit případnou kalafunu nebo pájecí roztok).

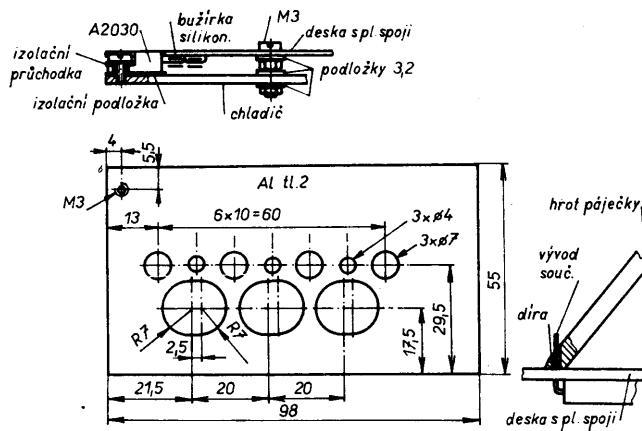
Pro určení středů otvorů slouží levé vylepšené plochy kruhů v kontaktních ploškách na spojové desce a pokud si nejsme jisti přesnou prací při vrtání, vrtáme díry menší a dopilujeme je podle skutečnosti za neustálého zkoušení na zdroji.

Pokud nepoužijeme doporučený zdroj, deska umožňuje připájet či přišroubovat napájecí zdírky pro externí zdroj, připájet nebo přišroubovat banánky pro přímé nasunutí do zdírek zdroje nebo přímo připájet vodiče. V tomto případě využijeme předleptaných pravých ploch kruhů v kontaktních ploškách (jako středy zdírek nebo banánek).

Při delších přívozech napájecího napětí je vhodné na desku připájet do místa přívodu blokovací kondenzátory 100 nF. Tyto kondenzátory nejsou nutné při doporučeném zdroji, deska je navržena tak, aby obvody byly stabilní i bez použití blokovacích kondenzátorů.

Některé výkonové zesilovače A2030 v použitém zapojení mají snahu kmitat s amplitudou asi 100 mV i přes použitý Boucherotův člen na výstupu ($R_{14} = 1 \Omega$ a $C_4 = 150 \text{ nF}$). Kapacitu kondenzátoru C_4 proto, nemáme-li možnost kontroly osciloskopem, volíme minimálně 220 nF.

Obvod A2030 je upevněn elektricky izolovaně (pomocí izolační průchody a izolační slídrové nebo teflonové podložky podle obr. 4 a obr. 16) na chladicí desku, zhotovenou podle obr. 4.



Obr. 16, 17. Chladicí deska a úprava hrotu páječky

Oválné otvory v chladicí desce nedovolují chyběně nasadit desku na svorky zdroje. Tato otvory musí být opět dosti přesně provedeny, nejlépe určíme jejich středy pomocí vlastní spojové desky, na níž označíme středy kružnic z předleptaných a vyvrtaných levých i pravých dír v kontaktních ploškách.

Vstupní zdírky nejlevnějšího provedení jsou přímo zapájeny do vyvrtaných dír v desce s plošnými spoji. Před zapájením je dokonale očistíme brusným papírem, zasuneme všechny zdírky, obrácenou desku položíme na rovný povrch a pájíme běžnou pistolovou páječkou. Zploštělé konce zdírek odštípneme (viz obr. 4).

U obvodu A2030 opatrně narovnáme vývody a znova je ohneme na druhou stranu podle obr. 4 a obr. 16. Na vývody navlékne me tenké silikonové bužírky, aby nemohl vzniknout zkrat. Můžeme také použít část slídrové nebo teflonové podložky pro odizolování plošných spojů. Je také možné umístit obvod shora a spojit jej s chladicí deskou kovovou kostkou rozměrů $15 \times 10 \times 5$ mm. Vhodná distanční vzdálenost mezi plošnými spoji a chladicí deskou je zajištěna dvěma podložkami pro šrouby M3 a maticí podle obr. 16. Protože pro upevnění chladicí desky stačí dva šrouby M3, může být třetí šroubek využit pro upevnění případné krycí desky ze strany součástek. Vzhledem k moderní tendenci, aby studenti viděli součástky obvodu ve skutečnosti a protože výškově přesahující zdírky chrání součástky před poškozením a na desce nelze snadno vnějším zásahem udělat zkrat, není tato krycí deska bezpodmínečně nutná (může být zhotovena např. i z tenkého organického skla).

Použitím chladicí s větší chladicí plochou a výkonnějšího zdroje napájecího napětí můžeme zvětšit proudový odběr z programovatelného zdroje až na velikost, danou mezními parametry obvodu A2030. Konstrukce desky umožňuje její vestavění na prakticky libovolný dostupný chladicí.

Pro vyvedení všech digitálních signálů z desky s plošnými spoji plochým vodičem je vhodné použít přímořezný konektor, určený pro 14vývodové IO, který je k dostání opět ve specializovaných prodejnách, např. v Praze u fy GM za asi 30 Kčs. Pak je nutné použít originální plochý vodič s roztečí drátů 1,25 mm, který lze koupit tamtéž za opět asi 30 Kčs i na dobírku. Toto provedení je na obr. 18 nahoře.

Je ovšem možné připájet přímo plochý vodič naši výroby za asi 11 Kčs do desky s plošnými spoji a zajistit proti vytržení spojů z plechu, podloženou kouskem ploché pryže a přišroubované ke spojové desce podle obr. 18 a obr. 25a. Pro spojení chladicí desky s deskou s plošnými spoji použijete střední díru (pro šroubek M3) mezi kanály Y a X.

Pro pájení na desce je vhodné upravit hrot páječky podle obr. 16, 17 jeho opilováním a vyvrtáním díry asi 0,9 mm. Práce s tímto hrotom je velice rychlá a vzhledná, hrotom nabereme z trubkové pásky cín do vyvrtané díry v hrotu, navléceme na vývod součástky a přitisknutím k pájecí ploše vytvoříme dokonalý spoj. Jako rezistory je možné použít destičkové typy WK.

Nastavení a seřízení jednotky

Ačkoliv toto zařízení nemá žádné nastavovací ani seřizovací prvky a nevyžaduje speciální přesné součástky (především rezistory do vstupních děličů), nelze obejít jeden ze zákonů G.F.W. Hegela – zákon dialektiky.

I toto jednoduché a levné zařízení musí mít nějakou vadu na úkor složitosti a ceny. Kompenzace této jednoduchosti je ukryta ve složitějším programovém vybavení a v nutnosti (pokud je zařízení osazeno doporučovanými rezistory typu TR 191,5 %) nastaví jednotku pomocí části XYZRM řidicího programu na obr. 14.

Jedná se o zjištění programových konstant RefNap, které jsou vlastně vstupními napětími pro jednotlivé rozsahy, odpovídající na výstupu převodníku A/D číslu 255 – tedy maximální kladné velikosti.

Pro nastavení použijeme zapojení podle obr. 11, tvořené regulovatelným zdrojem vstupního napětí a voltmetrem s přesností pokud možno lepší nebo srovnatelnou s nastavovaným zařízením. Jako zdroj regulovatelného napětí zvláště pro spodní rozsahy je vhodná např. plochá baterie s připojeným dvojitým potenciometrem (nebo dvěma potenciometry jednoduchými) s logaritmickým průběhem, kterým se lépe regulují malá napětí.

Obr. 18. Fotografie zařízení (viz 1. a 4. stranu obálky)

Obr. 14. XYZRM – program řízení a nastavení převodníků z obr. 5 a 13

```

uses DOS, CRT;
const
  RefNapetiZ01 = 0.193;
  RefNapetiZ1 = 1.16;
  RefNapetiY1 = 1.11;
  RefNapetiY10 = 12;
  RefNapetiX1 = 1.07;
  RefNapetiX10 = 9.78;
  RefNapetiX50 = 55.3;
  RefNapetiD00 = 121;
var
  Ch: char;
  kt,kx,kz,K,R,i,d,Vstup : byte;
  BX,BX50,BX10,BX1,BY10,BY1,BZ1,BZ01 : real;
  p : integer;
PROCEDURE ZhasniKursor;
var Regs: registers;
begin
  with Regs do
    begin
      AH := $01;
      CH := $20;
    end;
  Intr($10,Regs);
end;
PROCEDURE Aprox;
begin
  Delay(1);R := 0; Vstup := 0;
  for i := 7 down to 0 do
  begin
    Port[$3783] := R + 1 shl i;           {pro schéma na obr.4 a obr.5 je Delay(1)}
    Vstup := Port[$3793];
    if Vstup > 128 then R := R + 1 shl i;
  end;
  Port[$37A3]:=2; Port[$37B3]:=d;
end;
PROCEDURE v;{volba}
begin
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch#0 then
    begin
      Ch:=ReadKey;
      if Ch#79 then Halt;
      if Ch#77 then
      begin
        TextBackGround(LightGray); TextColor(0); gotoXY(1,1);
        writeln(' KANAL X [1 a2 50 V] KANAL Y [1 a2 10 V] KANAL Z');
        writeln(' [0..1 a2 1 V] D/A PŘEVODNIK ');
        TextBackGround(black);
        if p22 then p:=-1; inc(p);
      end;
    if Ch#75 then
    begin
      TextBackGround(LightGray); TextColor(0); gotoXY(1,1);
      writeln(' KANAL X [1 a2 50 V] KANAL Y [1 a2 10 V] KANAL Z');
      writeln(' [0..1 a2 1 V] D/A PŘEVODNIK ');
      TextBackGround(black);
      if p11 then p:=4; dec(p);
    end;
    if (Ch#72) and (kr=1) then
    begin
      if d < 255 then
      inc(d);
    end;
    if (Ch#80) and (kr=1) then
    begin
      if d > 0 then
      dec(d);
    end;
    if kr=1 then
    begin
      TextColor(15); gotoXY(65,2);
      writeln(' [ U ] = ',round((d-RefNapetiD00)*0.8)/10:2:1,' V ');
      Port[$3783]:=d; delay(100);
      if kt=1 then
      begin
        gotoXY(71,4);writeln('d = ',d,' ');
      end;
    end;
  end;
  end;
end;
PROCEDURE X;
begin
  v; TextColor(9);
  if keypressed then
  begin
    Ch:=ReadKey;
    if Ch#27 then kz:=0;
  end;
  Port[$37A3]:=10; Aprox; K:=A;
  if K=255 then K:=254; Port[$37A3]:=14; Aprox;
  if kt=1 then
  begin
    gotoXY(5,4);writeln(' AX1=',A,' ');
  end;
  BX1:=round(RefNapetiX1 *100*(A-K)/(255-K))/0.1;
  if (A=255) OR (A=0) then
  begin
    Port[$37A3]:=13; aprox;
    if kt=1 then
    begin
      gotoXY(5,5);writeln(' AX10=',A,' ');
    end;
    BX10:=round(RefNapetiX10 *100*(A-K)/(255-K))/100;
    if (A=255) OR (A=0) THEN
    begin
      Port[$37A3]:=12; aprox;
      if kt=1 then
      begin
        gotoXY(5,6);writeln(' AX50=',A,' ');
      end;
      if (A=255) OR (A=0) then
      begin
        gotoXY(1,2); write(' Ux větší než 50 V ');
      end;
      BX50:=round(RefNapetiX50 *10*(A-K)/(255-K))/10;
      gotoXY(1,2); write(' [ X50 ] = ',BX50:4:1,' [ V ] ');
      BX:=BX50;
    end;
  end;
  ELSE
  begin
    gotoXY(1,2); write(' [ X10 ] = ',BX10:4:1,' [ V ] ');
    BX:=BX10;
  end;
end;
PROCEDURE Y;
begin
  v; TextColor(12);
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch#27 then ky:=0;
  end;
  Port[$37A3]:=10; Aprox; K:=A;
  if K=255 then K:=254;
  Port[$37A3]:=9; Aprox;
  if kt=1 then
  begin
    gotoXY(27,4);writeln(' AY1=',A,' ');
  end;
  BY1:=round(RefNapetiY1 *100*(A-K)/(255-K))/0.1;
  if (A=255) OR (A=0) then
  begin
    Port[$37A3]:=8; aprox;
    if kt=1 then
    begin
      gotoXY(27,5);writeln(' AY10=',A,' ');
    end;
    BY10:=round(RefNapetiY10 *10*(A-K)/(255-K))/10;
    if (A=255) OR (A=0) THEN
    begin
      gotoXY(22,2); write(' Uy větší než 10 V ');
    end;
    else
    begin
      gotoXY(22,2); write(' [ Y10 ] = ',BY10:4:1,' [ V ] ');
      BY:=BY10;
    end;
  end;
  else
  begin
    gotoXY(22,2); write(' [ Y 1 ] = ',BY1:4:0,' [ mV ] ');
    BY:=BY1/1000;
  end;
end;
PROCEDURE Z;
begin
  v; TextColor(10);
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch#27 then kz:=0;
  end;
  Port[$37A3]:=10; Aprox; K:=A;
  if K=255 then K:=254; Port[$37A3]:=11; Aprox;
  if kt=1 then
  begin
    gotoXY(48,4);writeln(' AZ01=',A,' ');
  end;
  BZ01:=round(RefNapetiZ01 *1000*(A-K)/(255-K));
  if (A=255) OR (A=0) then
  begin
    Port[$37A3]:=15; aprox;
    if kt=1 then
    begin
      gotoXY(48,5);writeln(' AZ1=',A,' ');
    end;
    BZ1:=round(RefNapetiZ1 *100*(A-K)/(255-K))/0.1;
    if (A=255) OR (A=0) THEN
    begin
      gotoXY(43,2); write(' Uz větší než 1 V ');
    end;
    else
    begin
      gotoXY(43,2); writeln(' [ Z 1 ] = ',BZ1:4:0,' [ mV ] ');
      BZ:=BZ1/1000;
    end;
  end;
  else
  begin
    gotoXY(43,2); writeln(' [ Z01 ] = ',BZ01:4:0,' [ mV ] ');
    BZ:=BZ01/1000;
  end;
end;
PROCEDURE r;
begin
  v; TextBackGround(black); TextColor(15); gotoXY(65,2);
  writeln(' [ U ] = ',round((d-121)*0.8)/10:2:1,' V ');
end;
PROCEDURE Zx;
begin
  if Ch#13 then kx:=1;
  if Ch#27 then
  begin
    kx:=0; TextBackGround(black); gotoXY(1,2); write(' ');
    gotoXY(1,5); write(' ');
    gotoXY(1,8); write(' ');
  end;
  TextBackGround(black); TextColor(14); gotoXY(1,1);
  writeln(' KANAL X [1 a2 50 V]');
end;
PROCEDURE Zy;
begin
  if Ch#13 then ky:=1;
  if Ch#27 then
  begin
    ky:=0; TextBackGround(black); gotoXY(22,2); write(' ');
    gotoXY(22,4); write(' ');
    gotoXY(22,5); write(' ');
  end;
  TextBackGround(black); TextColor(14); gotoXY(22,1);
  writeln(' KANAL Y [1 a2 10 V]');
end;
PROCEDURE Zz;
begin
  if Ch#13 then kz:=1;
  if Ch#27 then
  begin
    kz:=0; TextBackGround(black); gotoXY(43,2); write(' ');
    gotoXY(43,4); write(' ');
    gotoXY(43,5); write(' ');
  end;
end;

```

```

TextBackGround(black); textColor(14); gotoXY(43,1);
write(' KANAL Z [0.1 až 1 V] ');
end;

PROCEDURE Zr;
begin
  if Ch=#13 then kr:=1;
  if Ch=#27 then
  begin
    kr:=0; TextBackGround(black); gotoXY(65,2); write(' ');
    gotoXY(65,4); writeln(' ');
  end;
  TextBackGround(black); textColor(14); gotoXY(65,1);
  write(' D/A PREVODNIK ');
end;

BEGIN
ZhasniKursor;
TextBackGround(0); ClrScr; d:=RefNapetiDA0; Port[$37A]:=2; Port[$37B]:=d;
kx:=#0;ky:=#0;kz:=#0;px:=#0;kt:=#0;
TextBackGround(black); textColor(15); gotoXY(5,10);
write('TEST pro nasaveni RD prevodniku ? AND stisknete [ A ] ');
write(' NE stiskete [ N ] ');
Ch:=ReadKey;
if (Ch=65) or (Ch=97) then kt:=1;
if (Ch=70) or (Ch=110) then kt:=0;
ClrScr; textbackground(LightGray); textColor(0);
write(' KANAL X [1 až 50 V] KANAL Y [1 až 10 V] ');
write(' KANAL Z [0.1 až 1 V] D/A PREVODNIK ');
gotoXY(1,0);
write('přerušení [END], volba kanálu [',chr(26),'], [',chr(27),'], [',chr(28),'], [',chr(29),'], ');
write('výber [ENTER], zrušení [ESC], fázení D/RL [',chr(24),'], [',chr(25),'], [',chr(26),'], ');
repeat
  ;
  if kx=1 then x;
  if ky=1 then y;
  if kz=1 then z;
  if kr=1 then r;
  if px=0 then zx;
  if px=1 then zy;
  if px=2 then zz;
  if px=3 then zr;
until false
END;

```

Obr. 15. XYZRG – program řízení převodníku z obr. 5 a 13 – graficky

```

uses CRT,GRAPH;

Const
  RefNapetiZ01 = 0.103;
  RefNapetiZ1 = 1.16;
  RefNapetiY1 = 1.11;
  RefNapetiY10 = 12;
  RefNapetiX1 = 1.07;
  RefNapetiX10 = 9.78;
  RefNapetiX50 = 55.3;
  RefNapetiDA0 = 121;
  BigFont = SmallFont;  (globální nastavení fontu)

Var
  Ch : char;
  vx,vx50,vy,vz,kx,ky,kz,K,R,i,d,Vstop : byte;
  SBX,SBY,SBZ,SBR,BR1,SBR1,BX,BX50,BX10,BX1,BY,BY10,BY1,BZ,BZ1,BZ01 : real;
  p ,Gd,Gm : integer;
  St : string[80];

```

```

PROCEDURE Aprox;
begin
  Delay(1);R := 0; Vstop := 0;
  for i := 7 downto 0 do
  begin
    Port[$37B] := R + 1 shl i;  (pro schéma na obr.4 a obr.5 je Delay(1))
    Delay(0);
    Vstop := Port[$37B];
    if Vstop > 128 then R := R + 1 shl i;
  end;
  Port[$37A]:=2; Port[$37B]:=d;
end;

PROCEDURE MR;
begin
  SetFillStyle(1,15);
  SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(500,17,' D/A PREVODNIK ');
  Bar3D(322,3,477,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(396,17,' KANAL Z [0.1 - 1 V]');
  Bar3D(161,3,316,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(235,17,' KANAL Y [1 - 10 V]');
  Bar3D(0,3,155,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(77,17,' KANAL X [1 - 50 V]');
  Bar3D(0,3,155,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(white);SetTextJustify(CenterText,CenterText);
  OutTextXY(235,17,' KANAL Z [0.1 - 1 V]');
end;

procedure Zx;
begin
  SetFillStyle(1,black);
  Bar3D(161,3,316,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(white);SetTextJustify(CenterText,CenterText);
  OutTextXY(235,17,' KANAL Z [0.1 - 1 V]');
end;

Procedure Zy;
begin
  SetFillStyle(1,black);
  Bar3D(322,3,477,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(white);SetTextJustify(CenterText,CenterText);
  OutTextXY(235,17,' KANAL Y [1 - 10 V]');
end;

Procedure Zz;
begin
  SetFillStyle(1,black);
  Bar3D(322,3,477,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(white);SetTextJustify(CenterText,CenterText);
  OutTextXY(235,17,' KANAL Z [0.1 - 1 V]');
end;

Procedure Zr;
begin
  SetFillStyle(1,black);
  Bar3D(322,3,477,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(white);SetTextJustify(CenterText,CenterText);
  OutTextXY(500,17,' D/A PREVODNIK ');
  Bar3D(0,3,155,30,2,true);
  SetColor(15);SetTextJustify(CenterText,CenterText);
  OutTextXY(500,17,' D/A PREVODNIK ');
  Bar3D(483,3,500,30,2,true);Bar3D(500,3,614,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(500,17,' D/A PREVODNIK ');
  OutTextXY(491,17,Char(24));OutTextXY(629,17,Char(25));
end;

procedure Gx;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(0,40,155,GetMaxY-40,2,true);
end;

procedure Gy;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(161,40,316,GetMaxY-40,2,true);
end;

procedure Gz;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(322,40,477,GetMaxY-40,2,true);
end;

procedure Gr;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(483,40,637,GetMaxY-40,2,true);
end;

procedure Hx;
begin
  SetColor(12);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  OutTextXY(0,190,' U > Umax ');
end;

procedure Hy;
begin
  SetColor(12);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  OutTextXY(160,190,' U > Umax ');
end;

procedure Hz;
begin
  SetColor(12);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  OutTextXY(320,190,' U > Umax ');
end;

procedure Mhx;
begin
  SetColor(0);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  OutTextXY(160,190,' U > Umax ');
end;

procedure Mhy;
begin
  SetColor(0);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  OutTextXY(320,190,' U > Umax ');
end;

procedure Mhz;
begin
  SetColor(0);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  OutTextXY(320,190,' U > Umax ');
end;

procedure V(volba);
begin
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch=#0 then
    begin
      Ch:=ReadKey;
      if Ch=77 then Halt;
      if Ch=77 then
      begin
        m;
        if p>2 then p:=-1; inc(p);
        if p=0 then zx;
        if p=1 then zy;
        if p=2 then zz;
        if p=3 then zr;
      end;
      if Ch=75 then
      begin
        m;
        if p<1 then p:=4; dec(p);
        if p=0 then zx;
        if p=1 then zy;
        if p=2 then zz;
        if p=3 then zr;
      end;
      if (Ch=72) and (kr=1) then
      begin
        if d < 255 then
        inc(d);Port[$37A]:=2; Port[$37B]:=d;delay(100);
      end;
      if (Ch=80) and (kr=1) then
      begin
        if d > 0 then
        dec(d);Port[$37A]:=2; Port[$37B]:=d;delay(100);
      end;
      if kr=1 then
      begin
        BR:=(round((d-RefNapetiDA0)*0.8)/10);
      end;
    end;
  end;
end;

procedure X;
begin
  v;
  if keypressed then
  begin
    Ch:=ReadKey;
    if Ch=27 then kx:=0;
  end;
  Port[$37A]:=10; Aprox; K:=R;
  if K>255 then K:=254; Port[$37A]:=14; Aprox;
  BX:=round(RefNapetiX1 * 1000*(R-K)/(255-K))/10;
  if (R>255) OR (R=0) then
  begin
    Port[$37A]:=13; Aprox; BX10:=round(RefNapetiX10 * 1000*(R-K)/(255-K))/100;
    if (R=255) OR (R=0) THEN
  end;
end;

```

```

begin
  Port[837A]:=12; approx;
  BX50:=round(RefNapeti50 *10*(R-K)/(255-K))/10; BX:=BX50;
  if abs(BX50) >50 then BX:=50;
  end;
  ELSE
    BX:=BX10;
  END
else
  BX:=BX1/1000;
end;

```

PROCEDURE Y;

```

begin
  v;
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch=#27 then ky:=0;
  end;
  Port[837A]:=10; Aprox; K:=R;
  if K>255 then K:=254;
  Port[837A]:=9; Aprox; BY1:=round(RefNapetiY1 *100*(R-K)/(255-K))/0.1;
  if (R=255) OR (R=0) then
  begin
    Port[837A]:=8; approx; BY10:=round(RefNapetiY10 *10*(R-K)/(255-K))/10;
    if abs(BY10)>8.0 then BY10:= 8.0;
    BY:=(BY10);
  end
else
  BY:=(BY1/1000);
end;

```

PROCEDURE Z;

```

begin
  v;
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch=#27 then kz:=0;
  end;
  mhz;Port[837A]:=10; Aprox; K:=R;
  if K>255 then K:=254; Port[837A]:=11; Aprox;
  BZ01:=round(RefNapetiZ01 *1000*(R-K)/(255-K));
  if (R=255) OR (R=0) then
  begin
    Port[837A]:=15; approx; BZ1:=round(RefNapetiZ1 *100*(R-K)/(255-K))/0.1;
    if abs(BZ1)>999 then BZ1:=999;
    BZ:=(BZ1);
  end
else
  BZ:=(BZ01);
end;

```

Procedure R;

```

begin
  v;
end;

```

PROCEDURE ZobrazNapetiX(var SBX,BX: real);

```

var St: String[40];
BEGIN
  if kx=0 then exit;
  if abs(BX) >= 50 then hx;
  if abs(BX) >= 0.999 then
  begin
    if vx=1 then gx;
    vx:=1;mhz;SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(7,1,8,1);
    SetTextJustify(LeftText,CenterText); SetColor(10); tbarva jednotek;
    if BX > 0 then OutTextXY(10,140,'+[V]');
    else
      begin
        SetColor(0); OutTextXY(10,140,'+') ;SetColor(10); OutTextXY(10,140,'-[V]');
      end;
    BX:=abs(BX);
    if BX=SBX then exit;
    SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(6,1,12,1);
    St := '' ; str(SBX:5:1, St); SetColor(black); OutTextXY(-15,230, St);
    str(BX:5:1, St); SetColor(14); tbarva údaje
    OutTextXY(-15,230, St); SBX := BX;
  end
else
  begin
    if ux=1 then
    begin
      SBX:=0;gx;
    end;
    vx:=0;mhz;SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(5,1,8,1);
    SetTextJustify(LeftText,CenterText); SetColor(10);
    if BX >= 0 then OutTextXY(10,140,'+[MV]');
    else
      begin
        SetColor(0); OutTextXY(10,140,'+') ;SetColor(10); OutTextXY(10,140,'-[MV]');
      end;
    BX:=abs(BX);
    if BX=SBX then exit;
    SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(8,1,12,1);
    St := '' ; str(SBX*1000:6:0, St); SetColor(black); OutTextXY(-10,230, St);
    str(BX*1000:6:0, St); SetColor(14); OutTextXY(-10,230, St);
    SBX := BX;
  end;
end;

```

PROCEDURE ZobrazNapetiY(var SBY,BY: real);

```

var St: String[40];
begin
  if ky=0 then exit;
  if abs(BY) >= 9.9 then hy;
  if abs(BY) >= 0.999 then
  begin
    if vy=0 then gy;
    vy:=1;mhy;SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(7,1,8,1);
    SetTextJustify(LeftText,CenterText); SetColor(10);
    if BY > 0 then OutTextXY(170,140,'+[V]');
    else
      begin
        SetColor(0); OutTextXY(170,140,'+') ;SetColor(10); OutTextXY(170,140,'-[V]');
      end;
    BY:=abs(RY);
    if BY=SBY then exit;
    SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(8,1,12,1);
    St := '' ; str(SBY:5:1, St); SetColor(black); OutTextXY(80,230, St);
    str(BY:5:1, St); SetColor(14); OutTextXY(80,230, St);
    SBY := BY;
  end
else
  begin
    if vy=1 then
    begin
      SBY:=0;gy;
    end;
    vy:=0;mhy;SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(5,1,8,1);
    SetTextJustify(LeftText,CenterText); SetColor(10);
    if BY >= 0 then OutTextXY(170,140,'+[MV]');
    else
      begin
        SetColor(0); OutTextXY(170,140,'+') ;SetColor(10); OutTextXY(170,140,'-[MV]');
      end;
  end;
end;

```

SetColor(0); OutTextXY(170,140,'+') ;SetColor(10); OutTextXY(170,140,'-[MV]');
end;

BY:=abs(BY);
if BY=SBY then exit;
SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(8,1,12,1);
St := '' ; str(SBY*1000:6:0, St); SetColor(black); OutTextXY(30,230, St);
str(BY*1000:6:0, St); SetColor(14); OutTextXY(30,230, St);
SBY := BY;
end;

PROCEDURE ZobrazNapetiZ(var SBZ,BZ: real);

```

var St: String[40];
BEGIN
  if kz=0 then exit;
  if abs(BZ) >=999 then hz;
  mhz;SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(5,1,8,1);
  SetTextJustify(LeftText,CenterText); SetColor(10);
  if BZ >= 0 then OutTextXY(335,140,'+[MV]');
  else
    begin
      SetTextJustify(LeftText,CenterText); SetColor(0);
      OutTextXY(335,140,'+') ;SetColor(10); OutTextXY(335,140,'-[MV]');
    end;
  BZ:=abs(BZ);
  if BZ=SBZ then exit;
  SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(8,1,12,1);
  SetTextJustify(LeftText,CenterText);
  St := '' ; str(SBZ:6:0, St); SetColor(black); OutTextXY(190,230, St);
  SetTextJustify(LeftText,CenterText);
  str(BZ:6:0, St); SetColor(14); OutTextXY(190,230, St);
  SBZ := BZ;
end;

```

PROCEDURE ZobrazNapetiR(var SBR,BR: real);

```

var St: String[40];
BEGIN
  if kr=0 then exit;
  SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(7,1,8,1);
  SetTextJustify(LeftText,CenterText); SetColor(10);
  if BR >= 0 then OutTextXY(480,140,'+[V]');
  else
    begin
      SetTextJustify(LeftText,CenterText);
      SetColor(0); OutTextXY(480,140,'+') ;SetColor(10); OutTextXY(480,140,'-[V]');
    end;
  BR1:=abs(BR);
  if BR1=SBR1 then exit;
  SetTextStyle(BigFont, HorizDir,10); SetUserCharSize(6,1,12,1);
  SetTextJustify(LeftText,CenterText);
  St := '' ; str(SBR1:2:1, St); SetColor(black); OutTextXY(515,230, St);
  str(BR1:2:1, St); SetColor(14); OutTextXY(515,230, St);
  SBR1 := BR1;
end;

```

(zpocatek programu)

BEGIN

```

  Gd := Detect; InitGraph(Gd, Em, '');
  (pozn. soubory s příponou .bgi (att.bgi,cga.bgi,egavga.bgi,herc.bgi musí být v aktuálním adresáři nebo musí být stanovená cesta např. c:\lang\tpas\bgi\)
  a data musí být v aktuálním adresáři fonty pro texty v grafickém režimu tj.
  sans.chr,trip.chr,goth.chr,litt.chr)
  if GraphResult <> gr then
  begin
    writeln('Nelez prepnout do grafiky.');
    Halt(10);
  end;
  ClearDevice;setbkcolor(0);ClrScr;d:=RefNapetiDA0; Port[837B]:=2; Port[837B]:=d;
  kx:=0;ky:=0;kz:=0;kr:=0;p:=0;SBX:=0;SBY:=0;SBZ:=0;SBR:=0;SBR1:=0.1;BR:=0;SBR1:=0.1;
  Mjgx:=zx;gy:=yz;gr:=
  SetFillStyle(1,15);
  SetColor(0);SetTextJustify(LeftText,CenterText);
  Bar3D(0,330,837,345,2,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  OutTextXY(10,337,' prerusení stiskem [END], volba kanalu [')
  OutTextXY(105,337,chr(28));OutTextXY(350,337,chr(27));
  OutTextXY(360,337,') , vyber [ENTER] , zrusení [ESC] ');

```

```

  repeat
    v;
    if p=0 then
      begin
        if Ch=#13 then kx:=1;
        if Ch=#27 then
          begin
            kx:=0;gx;
          end;
      END;
    if p=1 then
      begin
        if Ch=#13 then ky:=1;
        if Ch=#27 then
          begin
            ky:=0;gy;
          end;
      END;
    if p=2 then
      begin
        if Ch=#13 then kz:=1;
        if Ch=#27 then
          begin
            kz:=0;gz;
          end;
      END;
    if p=3 then
      begin
        if Ch=#13 then kr:=1;
        if Ch=#27 then
          begin
            kr:=0;gr;
          end;
      END;
    if kx=1 then
      begin
        x;zobrazNapetiX(SBX,BX);
      end;
    if ky=1 then
      begin
        y;zobrazNapetiY(SBY,BY);
      end;
    if kz=1 then
      begin
        z;zobrazNapetiZ(SBZ,BZ);
      end;
    if kr=1 then
      begin
        r;zobrazNapetiR(SBR,BR);
      end;
  until false;
  CloseGraph;
END.

```

Obr. 19. AUTOVOLT – program řízení jednokálového několikarozsahového převodníku A/D

```

program AUTOVOLT;{platí pro převodník bez obvodu MAB 311}
{platí pro relé 12 V !!!!!!!}
uses DOS, CRT;
const
  RefNapeti01 = 0.1;
  RefNapeti02 = 0.2;
  RefNapeti05 = 0.5;
  RefNapeti1 = 1;
  RefNapeti2 = 2;
  RefNapeti5 = 5;
  RefNapeti10 = 10;
  RefNapeti20 = 20;
  RefNapeti50 = 50;
  RefNapeti100 = 100;
  RefNapeti200 = 200;
  RefNapeti500 = 500;
var
  Ch : char;
  kt,xx,ky,kr,kz,K,A,i,d,Vstup,a10,a20 : byte;
  B,B01,B02,B05,B1,B2,B5,B10,B20,B50,B100,B200,B500 : real;
  p : integer;
PROCEDURE ZhasniKursor;
var Regs: registers;
begin
  with Regs do
    begin
      AH := $01;
      CH := $20;
    end;
  Intr($10,Regs);
end;
PROCEDURE Aprox;
begin
  Delay(1);A := 0; Vstup := 0;
  for i := 7 downto 0 do
  begin
    Port[$3781] := A + 1 shl i; { pro převodník z obr. 6 je Delay(0)}
    Vstup := Port[$3781];
    if Vstup > 120 then A := A + 1 shl i;
  end;
  Port[$3781]:=2; Port[$3781]:=d;
end;
PROCEDURE Konec;
begin
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch=#0 then
    begin
      Ch:=ReadKey;
      if Ch=#79 then
      begin
        textmode(3);Halt;
      end;
    end;
  end;
end;
PROCEDURE Test;
begin
  d:=122;Port[$37A1]:=2; Port[$3781]:=d;Delay(300);
  Port[$37A1]:=10; Aprox; K:=A;if K=255 then K:=254;
end;
PROCEDURE Zobraz;
begin
  gotoXY(5,2); writeln('U = ',B:6:3,' [V] ');
end;
PROCEDURE K01;
begin
  d:=252;Port[$37A1]:=2; Port[$3781]:=d; Port[$37A1]:=11; Aprox;
  B01:=round(RefNapeti01 *1000*(A-K)/(255-K));
  B:=B01/1000;
  if kt=1 then
  begin
    gotoXY(4,4);write(' A01=' ,A,' ') ;write(' [ 01 ] =',B01:5:0,' [mV] ');
  end;
  gotoXY(4,4);
end;
PROCEDURE K02;
begin
  d:=255; Port[$37A1]:=15; approx;
  B02:=round(RefNapeti02 *1000*(A-K)/(255-K));
  B:=B02/1000;
  if kt=1 then
  begin
    gotoXY(4,5);write(' A02=' ,A,' ') ;write(' [ 02 ] =',B02:5:0,' [mV] ');
  end;
  gotoXY(4,5);
end;
PROCEDURE K05;
begin
  d:=255; Port[$37A1]:=14; approx; CX1;
  B05:=round(RefNapeti05 *1000*(A-K)/(255-K));
  B:=B05/1000;
  if kt=1 then
  begin
    gotoXY(4,6);write(' A05=' ,A,' ') ;write(' [ 05 ] =',B05:5:0,' [mV] ');
  end;
  gotoXY(4,6);
end;
PROCEDURE K1;
begin
  d:=255; Port[$37A1]:=13; approx; CX10;
  B1:=round(RefNapeti1 *100*(A-K)/(255-K))/0.1;
  B:=B1/1000;
  if kt=1 then
  begin
    gotoXY(4,7);write(' A1=' ,A,' ') ;write(' [ 1 ] =',B1:5:0,' [mV] ');
  end;
  gotoXY(4,7);
end;
PROCEDURE K2;
begin
  d:=255; Port[$37A1]:=12; approx; CX10;
  B2:=round(RefNapeti2 *100*(A-K)/(255-K))/0.1;
  B:=B2/1000;
  if kt=1 then
  begin
    gotoXY(4,8);write(' A2=' ,A,' ') ;write(' [ 2 ] =',B2:6:3,' [V] ');
  end;
  gotoXY(4,8);
end;
PROCEDURE K5;
begin
  d:=255; Port[$37A1]:=9; approx; CY1;
  B5:=round(RefNapeti15 *40*(A-K)/(255-K))/40;
  B:=B5/40;
  if kt=1 then
  begin
    gotoXY(4,9);write(' A5=' ,A,' ') ;write(' [ 5 ] =',B5:6:3,' [V] ');
  end;
  gotoXY(4,9);
end;
PROCEDURE K10;
begin
  d:=255;Port[$37A1]:=8; approx;a10:=a; CY10;
  B10:=round(RefNapeti10 *10*(A-K)/(255-K))/10;
  B:=B10/10;
  if kt=1 then
  begin
    gotoXY(4,10);write(' A10=' ,A,' ') ;write(' [ 10 ] =',B10:6:3,' [V] ');
  end;
end;
PROCEDURE K20;
begin
  d:=122; Port[$37A1]:=15; approx; a20:=a;
  B20:=round(RefNapeti20 *10*(A-K)/(255-K))/10;
  B:=B20/10;
  if kt=1 then
  begin
    gotoXY(4,11);write(' A20=' ,A,' ') ;write(' [ 20 ] =',B20:6:3,' [V] ');
  end;
end;
PROCEDURE K50;
begin
  d:=122; Port[$37A1]:=14; approx;
  B50:=round(RefNapeti50*5*(A-K)/(255-K))/5;
  B:=B50/5;
  if kt=1 then
  begin
    gotoXY(4,12);write(' A50=' ,A,' ') ;write(' [ 50 ] =',B50:6:3,' [V] ');
  end;
end;
PROCEDURE K100;
begin
  d:=122; Port[$37A1]:=13; approx;
  B100:=round(RefNapeti100 *(A-K)/(255-K));
  B:=B100/100;
  if kt=1 then
  begin
    gotoXY(4,13);write(' A100=' ,A,' ') ;write(' [ 100 ] =',B100:6:3,' [V] ');
  end;
end;
PROCEDURE K200;
begin
  d:=122; Port[$37A1]:=12; approx;
  B200:=round(RefNapeti200 *(A-K)/(255-K));
  B:=B200/200;
  if kt=1 then
  begin
    gotoXY(4,14);write(' A200=' ,A,' ') ;write(' [ 200 ] =',B200:6:3,' [V] ');
  end;
end;
PROCEDURE K500;
begin
  d:=122; Port[$37A1]:=9; approx;
  B500:=round(RefNapeti500 *(A-K)/(255-K));
  B:=B500/500;
  if kt=1 then
  begin
    gotoXY(4,15);write(' A500=' ,A,' ') ;write(' [ 500 ] =',B500:6:3,' [V] ');
  end;
end;
PROCEDURE MER2;
begin
  K20;
  IF (A=255) OR (A=0) THEN K50
  else exit;
  IF (A=255) OR (A=0) THEN K100
  else exit;
  IF (A=255) OR (A=0) THEN K200
  else exit;
  IF (A=255) OR (A=0) THEN K500
  else exit;
  IF (A=255) OR (A=0) THEN
  begin
    gotoXY(5,2); write(' U větší než 500 V ')
  end;
  END;
PROCEDURE MER1;
begin
  K01;
  if(A=255) OR (A=0) then K02
  else exit;
  IF (A=255) OR (A=0) THEN K05
  else exit;
  IF (A=255) OR (A=0) THEN K1
  else exit;
  IF (A=255) OR (A=0) THEN K2
  else exit;
  IF (A=255) OR (A=0) THEN K5
  else exit;
  IF (A=255) OR (A=0) THEN K10
  else exit;
  BEGIN
    textmode(1);ZhasniKursor;A10:=122;A20:=122;
    d:=122;Port[$37A1]:=2; Port[$3781]:=d; kt:=0;
    textBackground(black);textColor(15);gotoXY(5,10);
    writeln('TEST pro nastavení AD převodníku ?');
    writeln('ANO stiskněte [ A ]'); write(' , NE stiskněte [ N ] ');
    Ch:=ReadKey;
    if (Ch=#65) or (Ch=#87) then kt:=1;
    if (Ch=#78) or (Ch=#110) then kt:=0;
    ClrScr;textColor(2);gotoXY(4,1);
    writeln('přerušení stiskem [END]');textColor(15);
    repeat
    test;d:=255;Port[$37A1]:=2; Port[$3781]:=d;Delay(200);
    repeat
    A10:=122;konec;MER1;zobraz;
    until (A10=255) or (A10=0);d:=122;Port[$37A1]:=2; Port[$3781]:=d;Delay(400);
    repeat
    konec;MER2;zobraz;
    until (A20>55) AND (A20<200);A20:=122;
    until false;textmode(3);
  END.

```

Spustíme program XYZRM a zvolíme režim pro TEST. V nabízeném menu zvolíme režim měření v příslušném kanálu, do kterého je převedeno vstupní napětí kladné polarity. Toto vstupní napětí měníme tak dlouho, až konstanta A, příslušná automaticky zvolenému rozsahu, dosáhne velikosti 255. Zapišeme změřenou velikost vstupního napětí a postup opakujeme pro další rozsah nebo kanál.

Změřené údaje na jednotlivých rozsazích dosadíme do odpovídajících konstant na počátku programu. Tento postup pro přesné nastavení je možné několikrát opakovat, až velikost napětí indikovaná na obrazovce souhlasí s údajem testovacího voltmetru.

Pro nastavení převodníku, osazeného komparátorem MAB311, je možné místo baterie použít i vestavěný převodník D/A ve funkci programovatelného zdroje napětí, jež velikost je v programu XYZRM nastavitelná kurzorovými klávesami. Číslo, které se objevuje na obrazovce, je přibližná velikost výstupního napětí získaná výpočtem a je ji možné v testovacím režimu uvést v souhlas se skutečně vytvořeným měřeným výstupním napětím pomocí programové konstanty RefNapDA.

Měření a vlastní uživatelské využití

Při vlastním měření pak již testovací režim nepoužíváme. Program XYZRM pracuje pouze v textovém režimu a je velice rychlý (velký počet měření za sekundu). Jeho nevýhodou jsou malé číslice zobrazující napětí. Je možné číslice trochu zvětšit jiným režimem obrazovky, jako je tomu v programu pro řízení vicerozsahového měřicího přístroje na obr. 19, ale největší využití je při grafickém režimu v programu XYZRG, z něhož je na obr. 20 kopie obrazovky.

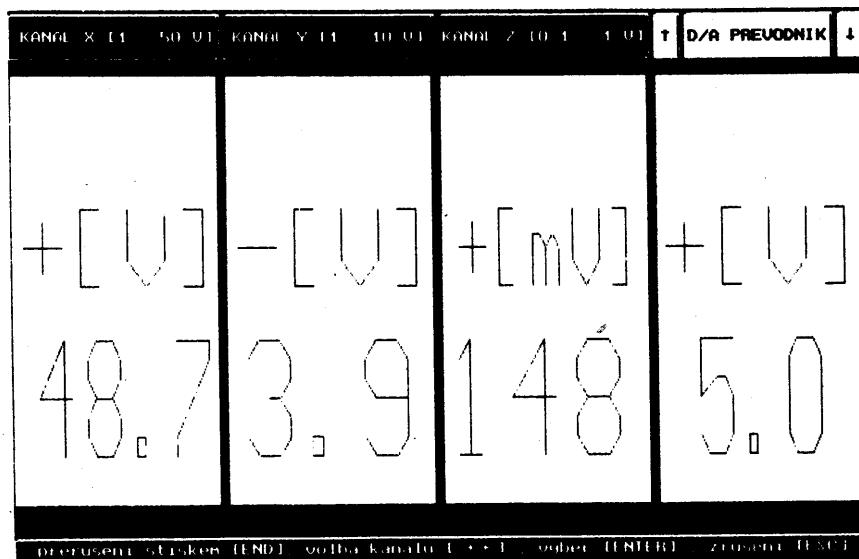
Tento program také vyžaduje pro přesná měření zadat programové konstanty RefNapDA, získané při nastavování. Kanály se volí kurzorovými klávesami VLEVO – VPRAVO, výběr klávesou ENTER, zrušení volby klávesou ESCAPE, řízení výstupního napětí převodníku D/A klávesami NAHORU–DOLŮ.

Tento program vyžaduje, aby v aktuálním adresáři byly soubory nutné pro činnost grafického režimu (soubory pro spolupráci s grafickými kartami – att.bgi, cga.bgi, egava.bgi, herc.bgi a fonty pro grafiku – sans.chr, trip.chr, litt.chr), nebo aby byla v příkazu InitGraph stanovena cesta k těmto souborům.

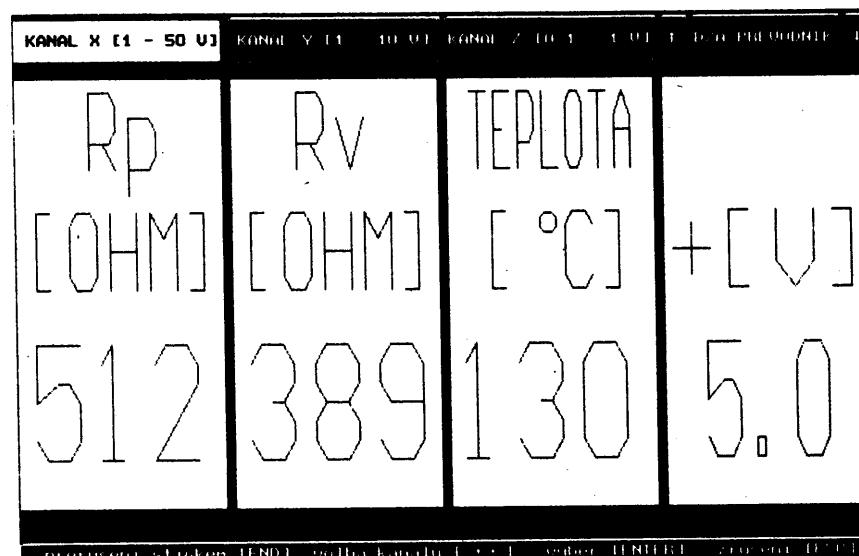
Program XYZRG lze upravit pro další tvůrčí práci na vlastním programovém vybavení, jak je možné vidět z kopie obrazovky na obr. 21, kde bylo využito dvou kanálů k měření závislosti odporu rezistorů z různých materiálů na teplotě a kde čidlo teploty bylo připojeno na další kanál. Výstupní napětí převodníku D/A bylo použito pro napájení měřicího přípravku.

Ve výpisu programu na obr. 15 jsou poznámky, umožňující další uživatelské změny.

To platí i o programu XYZRM, z něhož je možné použít programové rutiny PROCEDURE Aprox, PROCEDURE X nebo Y či Z, u nichž jsou vždy výstupem proměnné BX, BY, BZ, které udávají velikost napětí získanou převodem A/D ve [V]. Výstupní napětí převodníku D/A lze nastavit na libovolnou velikost od -10 V do +10 V ve 255 krocích po asi 80 mV příkazy Port[\$37A] := 2; Port



Obr. 20. Kopie obrazovky z programu XYZRG – tříkanálový voltmetr a programovatelný zdroj napětí



Obr. 21. Upravený program XYZRG – měření závislosti odporu na teplotě

[\$378] := d, kde d je číslo v rozsahu 0 až 255.

3kanálový osmibitový převodník A/D s automatickým přepínáním rozsahů, automatickou kompenzací posudu napěťové nesymetrie a posudu bipolární nuly, automatickou indikací polarity měřeného napětí, automatickou indikací překročení měřitelných parametrů a výkonový převodník D/A ve funkci programovatelného zdroje napětí do ± 10 V s delší dobou převodu

Tato alternativa realizovaná na téže desce s plošnými spoji (zapojení podle obr. 13) vznikne vynecháním komparátoru MAB311 (MAC111) a rezistoru R₁₁, naopak přibude rezistor R₁₂.

Jako komparátor je využita polovina některého z doporučených dvojitých operačních zesilovačů IO_{2,1}. Především vzhledem k dynamickým parametrům těchto běžných IO (doba zvětšování výstupního napětí)

musí být prodloužena doba mezi vysláním slova do obvodu MDAC08 a čtením stavu komparátoru v procedurách APROX o 1 ms, realizovaná příkazem DELAY(1). Tím se prodlouží celková doba převodu na asi 9 ms a zvětší se také zvlnění výstupního napětí převodníku D/A na asi 100 mV.

Menšího zvlnění lze dosáhnout zvětšením kapacity kondenzátorů C₂ a C₃ v analogové paměti. Je pak třeba prodloužit dobu převodu D/A.

Pokud budeme převodník D/A používat např. pouze ke spínání relé, jako je tomu v případě vicerozsahového měřicího přístroje, rozsvěcování žárovky, ovládání elektromagnetu či řízení rychlosti otáčení malého motorku, pak toto zvlnění nevadí. Při využití převodníku D/A např. k napájení měřicích můstků apod. se pak může použít další filtr až na výstupu převodníku D/A.

Zkrácení doby převodu

Dobu převodu u tohoto zapojení lze zkrátit použitím rychlejších a přesnějších dvojitých operačních zesilovačů, např. MAC412 nebo MAB412 či odpovídajících zahraničních typů.

Experimentálně lze měnit dobu zpoždění mezi zasláním slova do obvodu MDAC08 a čtením stavu komparátoru v procedurách APROX použitím tzv. prázdných příkazů, např. GOTOXY, WRITELN, WRITE a podobně, jejichž provedení trvá počítací určitou dobu, závislou na velikosti hodinového kmitočtu počítací. Správnou činnost ověřujeme pomocí programu XYZRM v režimu TEST. Měníme vstupní napětí v některém ze zvolených kanálů a sledujeme číslo získané převodem. To musí nabývat všech hodnot z intervalu 0 až 255.

Pro počítací XT byla takto ověřena minimální doba zpoždění pro bezchybný převod a pro obvod BO82 realizovaná příkazem WRITE zapsaným v proceduře APROX místo DELAY(0).

Pro počítací AT v režimu TURBO (12 MHz) je minimální časové zpoždění pro stejný obvod realizovatelné příkazem WRITELN.

Konstrukční provedení

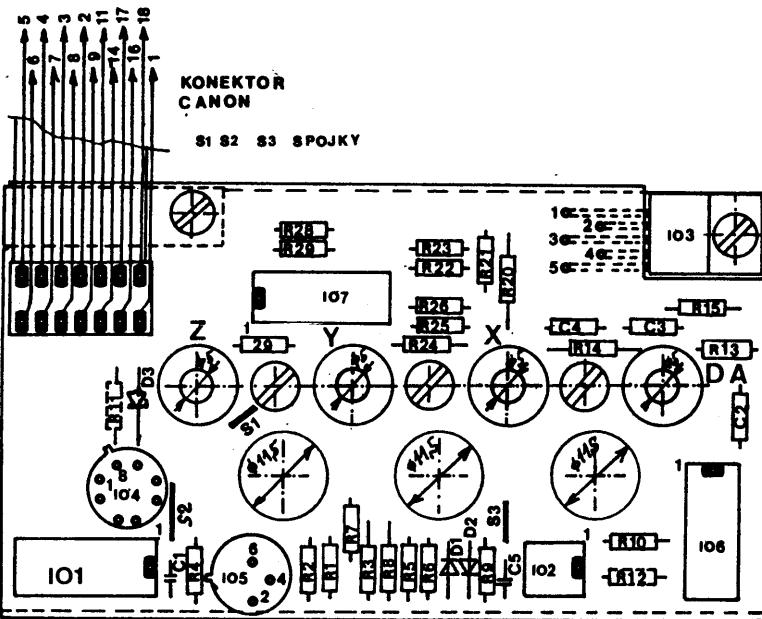
Pro osazení desky s plošnými spoji (obr. 22a) platí obr. 22, ve kterém se vynechá obvod IO4, rezistor R₁₁ a osadí se rezistor R₁₂. Tyto rezistory jsou nakresleny čárkovaně.

Pro výrobu i nastavení platí stejný postup jako v předchozím případě, stejně tak programové vybavení, ve kterém je v procedurách APROX provedena úprava časového zpoždění příkazem DELAY (1).

Jednokanálový osmibitový dvanáctirozahový převodník A/D s automatickým přepínáním rozsahů, automatickou kompenzací posudu napěťové nesymetrie a posudu bipolární nuly, automatickou indikací polarity měřeného napětí, automatickou indikací překročení měřitelných parametrů a výkonový převodník D/A ve funkci programovatelného zdroje napětí do ± 10 V

Zařízení představuje další variantu jednotky převodníků realizované na „univerzální“ desce s plošnými spoji. Úprava na obr. 23 zahrnuje především vstupní dělič, ochranné obvody a relé spínáno převodníkem D/A některým ze způsobů na obr. 24. V dalším budeme předpokládat zapojení podle obr. 24a.

Dále je vhodné zvětšit zisk vstupního zesilovače IO_{2,2} změnou odporu rezistoru R₅ na 120 k Ω , aby se kompenzoval vliv ochranných obvodů. Rozložení součástek na desce s plošnými spoji je na obr. 23a. Kromě úpravy odporu a zapojení rezistorů děliče je zapojen ještě další rezistor s odporem 10 M Ω , spínány programově relé, které je napájeno z převodníku D/A. Další rezistor 22 k Ω spolu s diodami D₄, D₅ chrání vstup



zvolí kanál S_1 . Toto napětí bude i na vstupu $IO_{2,2}$.

Multiplexer totiž sepne i tehdy, jestliže napětí na jeho vstupu dosáhne při napájení napětí ± 15 V velikost asi 11 V.

Proběhne měřicí cyklus, při kterém jsou postupně spínány jednotlivé rozsahy a testována velikost čísla, získaného převodem. Pokud je toto číslo 255 nebo 0, přepne se na výšší rozsah. V tomto případě bude číslo získané převodem trvale 255 a tak počítáč v dalším kroku nastaví napětí převodníku D/A na 0 V, relé rozepnou a napětí v bodě Z se zmenší vlivem předřadného rezistoru $10\text{ M}\Omega$ na asi jednu setinu, tj. na 220 mV. Opět proběhne měřicí cyklus a test na číslo získané převodem a přepínání rozsahů se zastaví na sepnutém S_6 a zobrazí se přepočtená a zaokrouhlená hodnota čísla získaného převodem.

Měřicí cyklus se neustále opakuje s rozpojenými kontakty relé až do té doby, kdy se napětí změní pod 10 V.

Pak relé opět sepnou a probíhá neustále měřicí cyklus se sepnutým relé až opět do té doby, než se vstupní napětí zvětší nad 10 V.

Tento systém řízení má tu výhodu, že je vždy maximálně využíván nejnižší a tedy i přesnější rozsah.

Pozor! Ochranný rezistor $22\text{ k}\Omega$ musí být minimálně na 15 W , protože průchodem proudu přes ochranné diody se na něm ztrácí při napětí (maximálním) 500 V činný výkon větší než 10 W . Dále musí být zachována konstrukční bezpečnost vstupní svorky, relé musí být schopno rozepnout maximální vstupní napětí (např. relé RP 210 na 12 V). Proto také je vstupní obvod (a ochranné obvody) mimo desku s plošnými spoji a rozsah měření je jen do 200 V . Zkušenější, vybavený potřebnými součástkami, mohou využít i rozsahy vyšších.

Vstupní odpor v provedení podle obr. 23 se mění z hodnoty $10\text{ M}\Omega$ na rozsazích vyšších než 10 V na asi $100\text{ k}\Omega$ na rozsazích nižších. Po dobu, než počítáč vyhodnotí stav pro rozepnutí relé (asi 1 ms), se vstupní odpor zmenší vlivem funkce ochranných obvodů asi na $20\text{ k}\Omega$.

Z hlediska chyby měření vlivem konečné velikosti proudu ochrannými diodami v závěrném směru a chyby způsobené svodovými proudy multiplexeru má největší vliv chyba způsobená svodovými proudy při zvýšené teplotě.

Za předpokladu, že vnitřní odpor zdroje měřeného napětí je značně menší než vstupní odpor, výstup multiplexeru pracuje do velkého odporu, odpor sepnutého kanálu je značně menší než ochranný odpor a při 7 využitých kanálech je chyba způsobená svodovými proudy za zvýšené teploty (55°) menší než velikost nejnižšího bitu. Blíže o vlivu ochranných obvodů na přesnost měření viz lit. [7].

Nejjednodušší převodník A/D na „univerzální“ desce s plošnými spoji

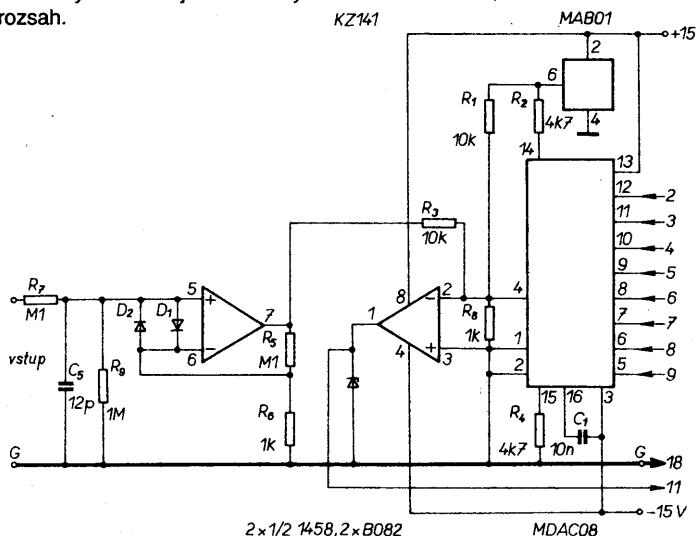
Schéma převodníku na obr. 25 a rozložení součástek na obr. 25a představuje nejjednodušší využití desky jako systému měřicího přístroje $\pm 100\text{ mV}$ se vstupním odporem $100\text{ k}\Omega$ a přesnosti 1 %. Za cenu tohoto zařízení (asi $150,-\text{ Kčs}$) nelze v současné době zakoupit ani analogové ručkové měřicí přístroje odpovídající přesnosti.

Programové vybavení AD4TEST a AD4 na obr. 26 a 27 slouží pro nastavení a vlastní měření. Doplňením tohoto zařízení o přepínač rozsahů a vstupní dělič se získá sice manuálně ovládaný, avšak levný měřicí přístroj.

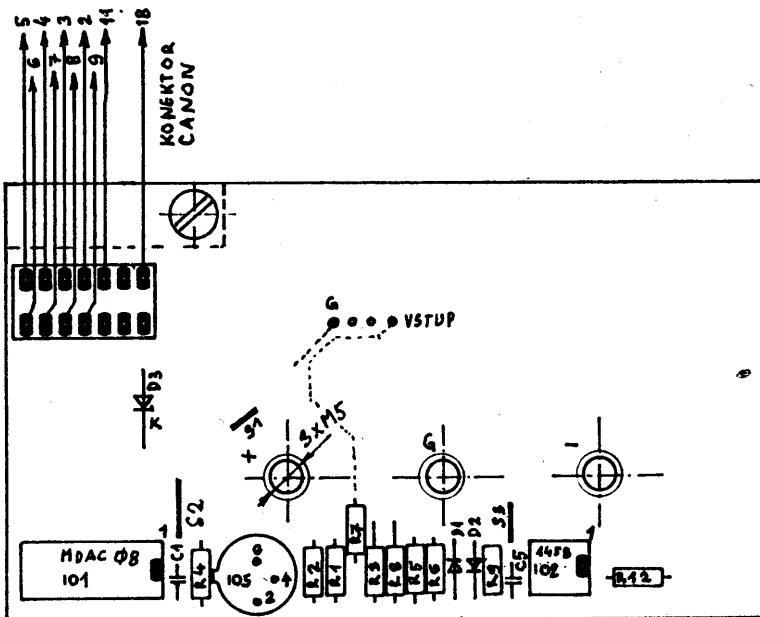
Postup nastavení a seřízení přístroje je uveden v kapitole o nastavení, úprava programového vybavení uživatelem s využitím programových rutin je v kapitole o měření a uživatelském využití.

Změnou rezistoru R_5 lze nastavit základní rozsah přístroje. Vzhledem k jednoduchosti jak obvodové, tak konstrukční jej lze doporučit k prvním experimentům i místo obvodů realizovaných na nepájivém kontaktním poli.

Pro napájení lze použít i zdroj $\pm 15\text{ V}$ „kalukačkového“ typu, tj. vestavěný v síťové výdlici, protože proudový odběr je malý.



Obr. 25. Schéma převodníku A/D s MA1458 a zesílením 100



Obr. 25a. Rozložení součástek převodníku z obr. 25 na desce s plošnými spoji

Obr. 26. AD4 – program řízení převodníku A/D z obr. 25

```
program AD4; { Program platí pro převodník na obr. 4 }

uses CRT, DOS;
const
  RefNapeti = 994; { zadejte napětí v mV }
  K = 122; { zadejte hodnotu K z první části testu }
var
  Ch : char;
  A,Vstup,i: byte;
  x:real;
  PROCEDURE ZhasniKursor;
  var Regs: registers;
  begin
    with Regs do
      begin
        AH := $01;CH := $20;
        Intr($10,Regs);
      end;
  end;
  begin
    ClrScr;ZhasniKursor;
    writeln('Přerušení měření stiskem libovolné klávesy');
    repeat
      A := 0; Vstup := 0;
      for i := 7 downto 0 do
        begin
          Port[$3780] := A + i shl 1;
          delay(1);
          Vstup := Port[$3791];
          if Vstup > 128 then A := A + 1 shl i;
        end;
      x:=round(RefNapeti *(A-K)/(255-K));gotoXY(1,3);
      writeln('napětí = ',x:4:0,' mV');
    until KeyPressed;
    Ch := ReadKey;
  end.
```

Obr. 27. AD4TEST – program nastavení převodníku z obr. 25

```

uses CRT;
var
  Ch : char;
  A,Vstup,i: byte;
begin
  ClrScr;
  writeln('Zkratujte vstupní svorky co nejkratším vodičem, vyčkejte cca 10 minut');
  writeln('a odečtete a zapíšte hodnotu A.');
  writeln('Qdstrante zkrat, pripojte regulovatelné vstupní napětí a voltmetr.');
  writeln('Zvýšujte vstupní napětí tak dlouho, až číslo na monitoru dosáhne 255');
  writeln('Zapište si toto vstupní napětí, přerušte program stiskem libovolné');
  writeln('klávesy. V programu AD4 zadejte toto napětí jako konstantu K');
  writeln('RefNapeti,hodnota A z první části testu zadejte jako konstantu K ');
  writeln('a program zaznamenejte. Dále volejte jen program AD4');
repeat
  A := 0;
  Vstup := 0;
  for i := 7 downto 0 do
  begin
    Port[$378] := A + 1 shl i;
    Delay(1);
    Vstup := Port[$379];
    if Vstup < 128 then A := A + 1 shl i;
  end;
  gotoXY(1,10);
  writeln('Hodnota A = ', A);
until KeyPressed;

Ch := ReadKey;
end.

```

Obr. 28. XYZRGST – program pro řízení převodníku A/D z obr. 13 (grafický), umožňující měřit stejnosměrná i střídavá napětí

```

uses CRT,GRAPH;
const
  RefNapetiZ01 =0.12;
  RefNapetiZ1 =1.219;
  RefNapetiY1 =1.12;
  RefNapetiY10 =12.2;
  RefNapetiX1 =1.1;
  RefNapetiX10 =10;
  RefNapetiX50 =55.5;
  RefNapetiDAO =120;
  BigFont = SmallFont;  {globální nastavení fontu}

var
  Ch : char;
  g_kxx1,kxx10,kxx50,vx,vx50,vy,vz,kx,ky,kr,kz,K,A,d,Vstup : byte;
  sx1,sx2,sy1,sy2,sz1,sz2 : byte;
  SBX,SBY,SBZ,SBR,BR,BRI,SBRI,BX,BX50,BX10,BX1,BY,BY10,BY1,BZ,BZ1,BZ01 : real;
  i,n,p,Gm : integer;
  St : string[80];
  xx1,xx10,xx50,yy1,yy10,zz1,zz01 : array[1..256]of integer;
  axx1,axx10,axx50,sxx1,sxx10,sx50:array[1..256]of real;
  ayy1,ayy10,azz1,azz01,azz1,yy10,szz1,szz01,szz1 : array[1..256]of real;

PROCEDURE Aprox1;
begin
  A := 0; Vstup := 0;
  for g := 7 downto 0 do
  begin
    Port[$378] := A + 1 shl g;
    Delay(0);
    Vstup := Port[$379];
    if Vstup > 128 then A := A + 1 shl g;
  end;
end;

PROCEDURE MA;
begin
  SetFillStyle(1,15);
  SetTextStyle(DefaultFont,horizDir,1);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(540,17,'/D/A PŘEVODNIK ');
  Bar3D(322,3,477,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(396,17,' KANAL Z [0.1 - 1 V]');
  Bar3D(161,3,316,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(235,17,' KANAL Y [1 - 10 V]');
  Bar3D(0,3,155,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(77,17,' KANAL X [1 - 50 V]');
  Bar3D(483,3,500,30,2,true);Bar3D(506,3,614,30,2,true);
  Bar3D(620,3,637,30,2,true);
  SetColor(0);SetTextJustify(CenterText,CenterText);
  OutTextXY(560,17,'/D/A PŘEVODNIK ');
  OutTextXY(491,17,Char(24));OutTextXY(629,17,Char(25));
end;

procedure Zx;
begin
  SetFillStyle(1,black);
  Bar3D(0,3,155,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(White);SetTextJustify(CenterText,CenterText);
  OutTextXY(77,17,' KANAL X [1 - 50 V]');
end;

Procedure Zy;
begin
  SetFillStyle(1,black);
  Bar3D(161,3,316,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(White);SetTextJustify(CenterText,CenterText);
  OutTextXY(235,17,' KANAL Y [1 - 10 V]');
end;

Procedure Zz;
begin
  SetFillStyle(1,black);
  Bar3D(322,3,477,30,2,true);SetTextStyle(DefaultFont,HorizDir,1);
  SetColor(White);SetTextJustify(CenterText,CenterText);
  OutTextXY(395,17,' KANAL Z [0.1 - 1 V]');
end;

procedure Zr;
begin
  SetFillStyle(1,black);
  Bar3D(483,3,500,30,2,true);Bar3D(506,3,614,30,2,true);
  Bar3D(620,3,637,30,2,true);
  SetColor(White);SetTextJustify(CenterText,CenterText);
  OutTextXY(560,17,'/D/A PŘEVODNIK ');
  OutTextXY(491,17,Char(24));OutTextXY(629,17,Char(25));
end;

procedure gx;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(0,40,155,GetMaxY-40,2,true);
end;

procedure gy;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(322,40,477,GetMaxY-40,2,true);
end;

procedure gz;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(322,40,477,GetMaxY-40,2,true);
end;

procedure gr;
begin
  SetColor(15); SetLineStyle(SolidLn,0,normWidth); SetFillStyle(SolidFill,0);
  Bar3D(483,40,637,GetMaxY-40,2,true);
end;

procedure hx;
begin
  SetColor(12);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  outTextXY(0,190,' U > Umax ');
end;

procedure hy;
begin
  SetColor(12);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  outTextXY(160,190,' U < Umax ');
end;

procedure hz;
begin
  SetColor(12);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  outTextXY(320,190,' U > Umax ');
end;

procedure mhx;
begin
  SetColor(0);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  outTextXY(0,190,' U > Umax ');
end;

procedure mhy;
begin
  SetColor(0);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  outTextXY(160,190,' U > Umax ');
end;

procedure mhz;
begin
  SetColor(0);SetTextStyle(DefaultFont,HorizDir,2);
  SetTextJustify(LeftText,CenterText);
  outTextXY(320,190,' U > Umax ');
end;

procedure v;
begin
  if KeyPressed then
  begin
    Ch:=ReadKey;
    if Ch#0 then
    begin
      Ch:=ReadKey;
      if Ch#79 then Halt;
      if Ch#77 then
      begin
        ma;
        if p>2 then p:=-1; inc(p);
        if p=0 then zx;
        if p=1 then zy;
        if p=2 then zz;
        if p=3 then zr;
      end;
      if Ch#75 then
      begin
        ma;
        if p<1 then p:=4; dec(p);
        if p=0 then zx;
        if p=1 then zy;
        if p=2 then zz;
        if p=3 then zr;
      end;
      if (Ch#72) and (kr=1) then
      begin
        if d< 255 then
          inc(d);Port[$37A]:=d; Port[$378]:=d;delay(100);
        if (Ch#80) and (kr=1) then
        begin
          if d > 0 then
            dec(d);Port[$37A]:=d; Port[$378]:=d;delay(100);
          if kr=1 then
          begin
            BR:=round((d-RefNapetiDAO)*0.8)/10;
          end
        end
      end
    end
  end;
end;

PROCEDURE XRSOST;
begin
  for i:=1 to (n+1) do
  begin
    sxx50[i]:=0;
  end;
  for i:=1 to n do
  begin
    sxx50[i+1]:=sxx50[i]+sqrt(axx50[i]);
    if (sxx50[i]=255) or (sxx50[i]=0) then
    begin
      BX:=50;exit;
    end;
    end;
    BX50:=round(10*sqrt(sxx50[n+1]/n))/10;
    BX:=BX50;
  end;

```

```

PROCEDURE XR50;
begin
PORT[37A]:=12;
for i:=1 to n do
begin
APROX1;xx50[i]:=a;
end;
PORT[37A]:=2;PORT[37B]:=d;
for i:=1 to n do
begin
if (sx1[i]=1) and (sx2=1) then
begin
xx50[i]:=RefNapetiX50*(xx50[i]-K)/(255-K);
sx50[i+1]:=sx50[i]+axx50[i];
if xx50[i]>K then sx1:=1;
if xx50[i]<K then sx2:=1;
end;
if (sx1=1) and (sx2=1) then
begin
XR50ST;exit;
end;
BX50:=round(sx50[n+1]/n*10)/10;BX:=BX50;
if abs(BX50)>50 then BX:=50;
BX:=BX50;
end;

PROCEDURE XR10ST;
begin
for i:=1 to (n+1) do
begin
sx10[i]:=0;
end;
for i:=1 to n do
begin
sx10[i+1]:=sx10[i]+sqr(axx10[i]);
end;
BX10:=round(100*sqr(sx10[n+1]/n))/100;BX:=BX10;
end;

PROCEDURE XR10;
begin
PORT[37A]:=13;
for i:=1 to n do
begin
APROX1;xx10[i]:=a;
if (a=255) or (a=0) then
begin
XR50;exit;
end;
PORT[37A]:=2;PORT[37B]:=d;
for i:=1 to n do
begin
axx10[i]:=RefNapetiX10*(xx10[i]-K)/(255-K);
sx10[i+1]:=sx10[i]+axx10[i];
if axx10[i]>0.2 then sx1:=1;
if axx10[i]<-0.2 then sx2:=1;
end;
if (sx1=1) and (sx2=1) then
begin
XR10ST;exit;
end;
BX10:=round(sx10[n+1]/n*10)/10;BX:=BX10;
end;

PROCEDURE XR1ST;
begin
for i:=1 to (n+1) do
begin
sx11[i]:=0;
end;
for i:=1 to n do
begin
sx11[i+1]:=sx11[i]+sqr(axx11[i]);
end;
BX1:=round(100*sqr(sx11[n+1]/n))/0.1;BX:=BX1/1000;
end;

PROCEDURE XR1;
begin
PORT[37A]:=14;
for i:=1 to n do
begin
APROX1;xx1[i]:=a;
if (a=255) or (a=0) then
begin
XR10;exit;
end;
PORT[37A]:=2;PORT[37B]:=d;
for i:=1 to n do
begin
axx1[i]:=RefNapetiX1*(xx1[i]-K)/(255-K);
sx1[i+1]:=sx1[i]+axx1[i];
if xx1[i]>K then sx1:=1;
if xx1[i]<K then sx2:=1;
end;
if (sx1=1) and (sx2=1) then
begin
XR1ST;exit;
end;
BX1:=round(100*sx1[n+1]/n)/0.1;BX:=BX1/1000;
end;

PROCEDURE YR10ST;
begin
for i:=1 to (n+1) do
begin
sy10[i]:=0;
end;
for i:=1 to n do
begin
sy10[i+1]:=sy10[i]+sqr(ayy10[i]);
if (yy10[i]=255) or (yy10[i]=0) then
begin
BY:=9.9;exit;
end;
end;
BY10:=round(100*sqr(sy10[n+1]/n))/100;
BY:=BY10;
end;

PROCEDURE YR10;
begin
PORT[37A]:=15;
for i:=1 to n do
begin
APROX1;yy10[i]:=a;
end;
PORT[37A]:=2;PORT[37B]:=d;
for i:=1 to n do
begin
ayy10[i]:=RefNapetiY10*(yy10[i]-K)/(255-K);
sy10[i+1]:=sy10[i]+ayy10[i];
if yy10[i]>K then sy1:=1;
if yy10[i]<K then sy2:=1;
end;

```

BZ1:=round(100*szz1[n+1]/n -0.4)/0.1;
if abs(BZ1)>999 then BZ1:=999;
BZ:=BZ1;
end;

```

PROCEDURE YR1ST;
begin
for i:=1 to (n+1) do
begin
szz0[i]:=0;
end;
for i:=1 to n do
begin
szz0[i+1]:=szz0[i]+sqr(azz0[i]);
end;
BZ01:=round(1000*sqrt(szz0[n+1]/n));BZ:=(BZ01);
end;
```

```

PROCEDURE YR1;
begin
PORT[37A]:=11;DELAY(1);
for i:=1 to n do
begin
APROX1;zz0[i]:=a;
if (a=255) or (a=0) then
begin
ZR1;exit;
end;
end;
PORT[37A]:=2;PORT[37B]:=d;
for i:=1 to n do
begin
ayy1[i]:=RefNapetiY1*(yy1[i]-K)/(255-K);
sy1[i+1]:=sy1[i]+ayy1[i];
if yy1[i]>K then sy1:=1;
if yy1[i]<K then sy2:=1;
end;
if (sy1=1) and (sy2=1) then
begin
ZR1ST;exit;
end;
BZ01:=round(1000*szz01[n+1]/n);BZ:=BZ01;
end;
```

```

PROCEDURE X;
begin
if keypressed then
begin
Ch:=ReadKey;
if Ch=>27 then kz:=0;
end;
Port[37A]:=10;DELAY(1);Aproxi;K:=A;
if K=255 then K:=254;
XR1;
end;
```

```

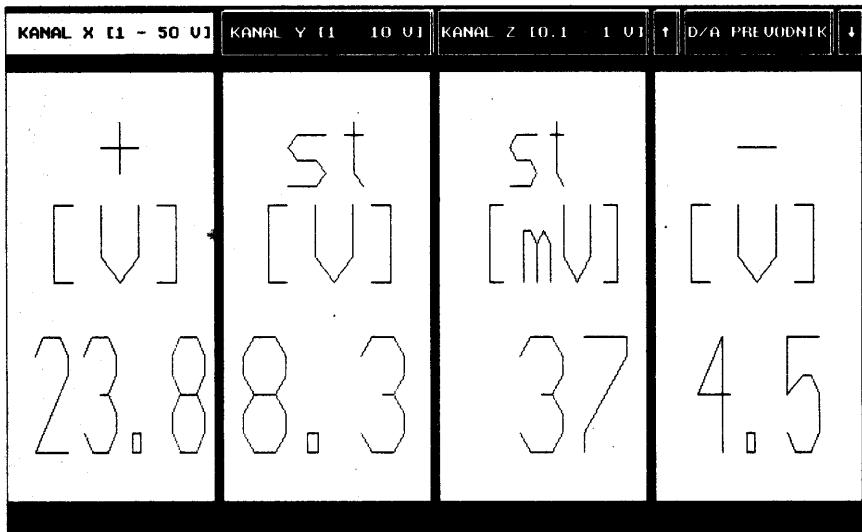
PROCEDURE Y;
begin
v;
if KeyPressed then
begin
Ch:=ReadKey;
if Ch=>27 then ky:=0;
end;
Port[37A]:=10;Delay(1);Aproxi;K:=A;
if K=255 then K:=254;
YR1;
end;
```

```

PROCEDURE Z;
begin
v;
if KeyPressed then
begin
Ch:=ReadKey;
if Ch=>27 then kz:=0;
end;
mhz;Port[37A]:=10;Delay(1);Aproxi;K:=A;
if K=255 then K:=254;
ZR01;
end;
```

```

PROCEDURE R;
begin
v;
end;
```



Obr. 29. Kopie obrazovky z programu XYZRGST


```

begin
  y; ZobrazNapetiY(SBV, BY);
  end;
  if kz=1 then
  begin
    z; ZobrazNapetiZ(SBZ, BZ);
  end;
  if kz=1 then
  begin
    r; ZobrazNapetiR(SBR, BR);
  end;
  until false;
CloseGraph;
END.

```

3kanálový osmibitový převodník A/D s automatickým přepínáním rozsahů, automatickou kompenzací posudu napětí a posudu bipolární nuly, automatickou indikací polarity měřeného napětí, automatickým přepínáním při měření stejnosměrného i střídavého napětí, automatickou indikací měření střídavého nebo stejnosměrného napětí, automatickou indikací překročení měřitelných parametrů a výkonový převodník D/A jako programovatelný zdroj napětí do ± 10 V

Programové vybavení uvedené na obr. 28 umožnuje využití převodníku z obr. 13 pro měření nejen stejnosměrných napětí, ale i pro měření střídavých napětí síťového kmitočtu (50 Hz), popř. v rozsahu 20 Hz až asi 500 Hz. Program zajišťuje rovněž číslicovou filtrace při měření stejnosměrných napětí. Uvedené programové vybavení umožnuje, aby při připojení libovolného napětí (v rozsahu dovolených velikostí, tj. 100 V ss nebo 65 V st hranice bezpečných napětí) ať stejnosměrného libovolné polarity, nebo střídavého, se na obrazovce objevil vždy správný údaj napětí včetně polarity (měření stejnosměrného napětí), nebo údaj velikosti střídavého napětí s písmeny st v místě zobrazování znaménka u ss napětí (obr. 29), případně indikace překročení měřitelných parametrů.

Zobrazovaný údaj v případě měření střídavého napětí je skutečná efektivní hodnota, získaná výpočtem z definice z naměřených vzorků. Umožnuje tedy měřit skutečnou efektivní hodnotu elektrických veličin i neharmonického (nesinusového) průběhu, jaké se mohou i ve školní či amatérské praxi vyskytnout poměrně často, např. při měření na přesycených magnetických obvodech, nebo při zkresleném průběhu výstupního signálu u přebuzených zesilovačů nebo u zesilovačů se špatně nastaveným pracovním bodem.

Počet vzorků je možné volit v programu XYZRGST volbou n , které je uvedeno hned na počátku hlavního programu po inicializaci grafiky. Experimentálně byl ověřen vhodný počet vzorků $n = 180$ pro měření napětí o kmitočtu 50 Hz až 500 Hz, který platí jak pro počítače AT, tak XT.

Ačkoliv by vzhledem k době převodu včetně uložení do paměti a délce periody napětí 50 Hz postačovalo asi 50 vzorků, způsoboval tento minimální počet vzorků blikání poslední číslice, tj. naměřený údaj by byl při každé periodě vzorkován jiný, i když v rozsahu dovolené nepřesnosti 1 % z údaje.

Počet vzorků $n=180$, uložený standardně v programu, umožnuje při jednokanálovém režimu asi 3 měření za sekundu pro počítače AT s kmitočtem hodin 12 MHz, což odpovídá počtu měření převodníky integračního typu, používanými v číslicových voltmetrech.

Při měření střídavých napětí se mění rozsahy v jednotlivých kanálech v poměru $1/\sqrt{2}$, tj. na $U_{ef} = 36$ V v kanálu X, 7 V v kanálu Y a 0,7 V v kanálu Z.

Měřicích procedur APROX1, X, Z, XR1, XR1ST, XR10, XR10ST, XR50, XR50ST, YR1ST, YR10, YR10ST, ZR01, ZR01ST, ZR1, ZR1ST je možné využít i pro tvorbu vlastních uživatelských programů např. v úpravě programu pro řízení vícerozsažového měřicího převodníku na obr. 23. Uvedené procedury odevzdávají proměnné BX, BY, BZ vhodné k dalšímu programovému zpracování.

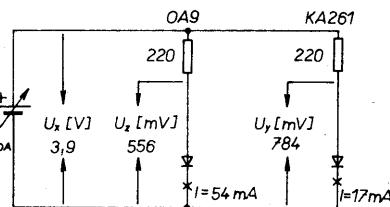
Náměty pro použití desky převodníků

1. Demonstrační experiment – dělič napětí

Vstupní napětí z D/A se měří jedním kanálem, druhým kanálem se měří napětí na výstupu děliče – použije se program XYZRG, nebo se vytvoří vlastní uživatelský program, při němž je na obrazovce znázorněn dělič, okamžité hodnoty napětí se zobrazují jednak v místech, v nichž je lze naměřit, jednak v teoretickém vzorci pro výpočet výstupního napětí. Možno použít pro nezatížený i zatížený dělič napětí. Možno zobrazit závislost U_2 na U_1 .

2, 3. Voltampérové charakteristiky germaniové a křemíkové diody.

Experiment využívá buď programu XYZRG, při němž jednotlivé kanály měří napětí na diodách v propustném směru a naopak převodníku D/A, které se mění ručně a přivádí na jednoduchý měřicí přípravek podle obr. 30a, nebo se vytvoří vlastní uživatelský program, při němž se pro demonstrační experiment napětí D/A zadává programově, měří se napětí na diodách pro každý krok, vypočítávají se ze známých odporek rezistorů a úbytků na diodách proudy a mohou se rovnou na obrazovce kreslit voltampérové charakteristiky. Pro laboratorní měření studenti mohou zadávat napětí ručně,



Obr. 30. Měření charakteristik polovodičových součástek, diod a tranzistorů – viz 4. strana obálky

na obrazovce jsou indikovány okamžité hodnoty napětí a proudů, po skončení měření se vytisknou charakteristiky (pro každého studenta), v nichž doma studenti např. mohou určovat v zadaném pracovním bodě stejnosměrné a dynamické odpory apod. Na obr. 30b je příklad takové úlohy měřené ještě pomocí počítače IQ 151. Podobně lze vytvořit programy pro měření např. charakteristik termistoru, variátoru a jiných nelineárních prvků, nebo přímo užit program XYZRG.

4. Ověření Ohmova zákona na jednoduchých obvodech.

Možno využít upravený program XYZRG, při němž jeden z kanálů indikuje proud měřením úbytku napětí na rezistoru malým odporem známé velikosti, nebo počítač vypočte proud z rozdílu úbytků napětí.

5. K jednotlivým kanálům je možné připojit i čidlo teploty, tlaku, fotodiodu, fotoodpor, sluneční baterii apod.

Velmi jednoduché čidlo teploty lze zhotovit z termistoru zapojeného v sérii s rezistorem

Obr. 31. Příklad měření výstupních charakteristik tranzistoru (viz 4. strana obálky)

Obr. 32. Zpracování změřených charakteristik tranzistoru (viz 3. strana obálky)

a napájené z převodníku D/A. Počítač měří napětí na termistoru úměrné teplotě a může provádět i početní linearizaci.

Podobně lze postupovat i pro jiná čidla.

6. Převodník D/A lze použít k napájení např. malé žárovky (a měřit osvětlení jedním kanálem převodníku A/D), elektromagnetu, topné šroubovice i k řízení malého motorku (je vhodné použítý motorek doplnit ochrannými a odrušovacími obvody) nebo k ovládání napěťových řízených zesilovačů či generátorů.

Převodník D/A lze použít i ke generování požadovaného průběhu výstupního napětí v závislosti na čase. Vzhledem k časové konstantě analogové paměti musí být mezi jednotlivými kroky časová prodleva minimálně 100 ms.

7. Pro jednotku bude vypracován program, který umožní její využití jako automatického digitálního osciloskopu. Takový program již byl ověřen na počítači IQ.

8. Další připravované programové vybavení umožní měřit efektivní hodnoty harmonického napětí bez použití lineárního usměrňovače a přímo měřit kmitočet v rozsahu akustických kmitočtů. Je možné ovšem připojit i lineární usměrňovač podle některého z již uveřejněných návodů, který může být spinán relé, řízeným převodníkem D/A např. podle obr. 24. Podobně může být připojen i převodník kmitočet-napětí a úpravou programu zobrazen digitálně měřený kmitočet na obrazovce.

9. Na obr. 31 je zobrazeno využití zařízení k měření výstupních charakteristik tranzistoru. Vstupní napětí určující proud do báze nastavují studenti ručně a udržují proud konstantní. Počítač měří toto vstupní napětí, napětí mezi bází a emitem a napětí mezi kolektorem a emitorem. Počítač programově zvětšuje prostřednictvím převodníku D/A napětí mezi kolektorem a emitorem ve 127 krocích a pro každý krok jsou měřena všechna tři napětí. Z naměřených údajů počítač vypočte proud báze i proud kolektoru a současně kreslí v souřadném systému $I_k = f(U_{ce})$ příslušné charakteristiky, takže studenti při měření vidí bezprostředně následky svého zásahu do experimentu. Zároveň se píše na obrazovku napětí a proudy do míst, v nichž je lze v uvedeném schématu naměřit (pro zvětšení didaktické účinnosti experimentu). Místo anonymních údajů běžných ručkových přístrojů při dosud běžném měření, je pro některé studenty a zvláště studenty s odrážetí těžíšem vztahem k elektronice použitá metoda přece jen didakticky názornější a účinnější.

Po odměření počítač vytiskne naměřené charakteristiky pro každého studenta a předkreslí i stupnice proudu báze. Doma pak studenti odvozují proudovou převodní charakteristiku, zakreslují zatěžovací přímkou pro zadaný kolektorový odpor, odvozují pracovní proudovou převodní charakteristiku a konstruují graficky průběh výstupního napětí pro maximální rozkmit proudu báze (viz obr. 32).

Pro vysvětlení principu funkce jednostupňového tranzistorového zesilovače je používán demonstrační experiment řízený počítačem (ještě IQ 151, bude upraven pro popisovanou jednotku).

10. Připravovaný program „automatický osciloskop“ umožní měřit i časové závislosti jako nabíjení a vybíjení kondenzátoru, zobrazení odezvu na skokové změny realizované převodníkem D/A apod.

Specifikace součástek

Integrované obvody

IO ₁	MDAC08 – libovolný typ podle požadované přesnosti
IO ₂	MA1458, B082, TL072, MAB412, MAC412
IO ₃	A2030V
IO ₄	MAB311, MAC111
IO ₅	MAB01, MAC01
IO ₆	MAB24E, MAC24
IO ₇	MAB08F, MAB08E, MAC08

Diody

D ₁ , D ₂	KA261 nebo libovolný typ rychlé malé diody
D ₃	KZ141, KZ260/4V7

Rezistory TR 191

R ₇ , R ₈	1 kΩ
R ₁₂ , R ₁₁	4,7 kΩ
R ₁₃	10 kΩ
R ₉ , R ₁₅	1 MΩ
R ₁₄	1 Ω /0,25 W

Obr. 33. AD2B – program řízení převodníku A/D z obr. 6 a 9, BASIC

```
0 CLS: K=10:PRINT "PRERUSENÍ MERENÍ STISKEM [CTRL]+[PAUSE]"
10 A=0:V=0
20 FOR I=7 TO 0 STEP (-1)
30 OUT &H378,A+2^I:PRINT
40 V=INP (&H379)
50 IF V> 128 THEN A=A+2^I
60 NEXT I
70 LOCATE 10,10,0
80 PRINT A/255*K"("V"
90 GOTO 10
0k PRINT A
110 GOTO 10
```

Obr. 34. AD2P – program řízení převodníků A/D z obr. 6 a 9, PASCAL

```
uses CRT,DOS;
const
  RefNapeti = 9.99; {zadejte napeti ve V}
var
  Ch : char;
  A,Vstup,i: byte;

PROCEDURE Zhasnikursor;
var Regs: registers;
begin
  with Regs do
    begin
      AH := $01;CH := $20;
    end;
  Intr($10,Regs);
end;

Procedure APPROX;
begin
  A := 0; Vstup := 0;
  for i := 7 downto 0 do
  begin
    Port[$378] := A + 1 shl i;
    delay(1);
    Vstup := Port[$379];
    if Vstup > 128 then A := A + 1 shl i;
  end;
end;

begin
  ClrScr;Zhasnikursor;
  writeln('Přerušení měření stiskem libovolné klávesy');
  repeat
    APPROX;gotoXY(1,3);
    writeln('napětí = ', round(RefNapeti *255/2.55 )/100 : 5:1,' V');
    until KeyPressed;
  Ch := ReadKey;
end.
```

Rezistory TR 191 nebo levné s tolerancí 1 % s kovovou vrstvou 0,25 W

R ₁ , R ₃ , R ₁₀ , R ₂₁ , R ₂₅ , R ₂₈	10 kΩ
R ₂ , R ₄	4,7 kΩ
R ₅ , R ₂₀ , R ₂₄ , R ₂₇ , R ₂₉	100 kΩ
R ₆ , R ₂₂ , R ₂₆	1 kΩ
R ₂₃	220 Ω

Pro jednokanálovou variantu na obr. 23

R₂₀ = 100 kΩ, R₂₁ = 56 kΩ, R₂₂ = 33 kΩ, R₂₃ = 12 kΩ, R₂₄ = 5,6 kΩ, R₂₅ = 3,3 kΩ, R₂₆ R₂₇ = 1,2 kΩ, R₅ = 0,12 MΩ, D₄, D₅ = KA263

Kondenzátory

C ₁	10 nF co nejmenší, keramický
C ₂ , C ₃ , C ₄	330 až 150 nF co nejmenší (viz text)
C ₅	12 pF co nejmenší, keramický

Konstrukční díly

plochý 14žilový vodič o Ø žily 0,15 mm, délka max. 1 m (v případě použití samořezného konektoru pro vývod z desky musí být rozteč žil 1,254 mm, viz text)
 konektor pro připojení tiskárny CANON 25 kontaktů (zapojení na obr. 1)
 kryt na konektor CANON
 14pólový „samořezný“ konektor pro IO-vývod z desky (viz text)
 chladič – deska z hliníkového plechu tl. = 2 mm
 izolační průchodka pro IO₃
 izolační podložka pro IO₃
 4 ks izolovaná zdířka obyčejná

Zkušenosti z provozu a možnosti úpravy jednotky

Doporučené úpravy se týkají především rozsahů a počtu kanálů převodníku A/D. Pro většinu prakticky uskutečňovaných měření

v amatérské praxi, na školách i v kroužcích mládeže se vystačí s rozsahem do 100 V (i s ohledem na velikost tzv. bezpečného napětí). Pro jednokanálové provedení je pak vhodné navrhnut vstupní dělič pro 0,1-0,3-1-3-10-30-100 V a nemusí se používat relé napájené z převodníku D/A, který se pak může použít k jiným účelům.

Jinou možností při zachování počtu kanálů je upravit rozsahy v kanálu Z na 0,1-0,3-1 V, v kanálu Y na 3-10 a v kanálu Z na 30-100 V. Po zvětšení odporu všech rezistorů děliče asi 3 až 5krát je pak možné spojovat kanály paralelně a tak upravovat rozsahy a počty kanálů, aníž by se zmenšil vstupní odpor pod asi 100 kΩ.

Nezapojením rezistoru R₁ a úpravou odporu rezistorů R₃ a R₁₀ se získá unipolární provoz převodníku jak A/D tak D/A.

Při použití komparátoru MAB311 je možné zmenšit kapacitu kondenzátorů v analogové paměti. Kapacita závisí na době převodu A/D a ta je závislá na hodinovém kmitočtu počítače. Nejvhodnější je ověřovat ji experimentálně s ohledem na dovolené zvlnění. Při používání programu XYZRGST pro měření stejnosměrných i střídavých veličin převodníkem z obr. 13 však kapacity kondenzátorů v analogové paměti převodníku D/A neměníme.

Převodník D/A v jednotce lze použít i pro spínání relé např. 5 V. Je však vhodné chránit relé Zenerovou diodou, protože přestože při zapnutí zdroje a připojeném zařízení k počítači je na výstupu převodníku D/A 0 V, může se při delším čase (než se spustí program) naintegrovat výstupní napětí na velikost větší než 5 V. Jakmile se spustí libovolný z řidicích programů, na výstupu se okamžitě nastaví výchozí hodnota 0 V. Na

Obr. 35. AD2BTEST – program nastavení převodníků A/D z obr. 6 a 9, BASIC

```
0 CLS :LOCATE 1,10,0:PRINT "PRERUSENÍ MERENÍ STISKEM [CTRL]+[PAUSE]"
5 LOCATE 3,1,0:PRINT "ZVÝŠUJTE VSTUPNÍ NAPĚTI AŽ ČÍSLO NA MONITORU BUDE 255"
6 LOCATE 4,1,0:PRINT "ZAPÍSETE SI TOTO VSTUPNÍ NAPĚTI A ZADEJTE JEJ KONSTANTU U V PROGRAMU AD2B"
10 A=0:V=0
20 FOR I=7 TO 0 STEP (-1)
30 OUT &H378,A+2^I
40 V=INP (&H379)
50 IF V> 128 THEN A=A+2^I
60 NEXT I
70 LOCATE 10,10,0
80 PRINT A
90 GOTO 10
```

Obr. 36. Program řízení převodníků A/D z obr. 6 a 9, PASCAL

```
uses CRT,DOS;
const
  RefNapeti = 9.99; {zadejte napeti ve V}
var
  Ch : char;
  A,Vstup,i: byte;

Procedure Zhasnikursor;
var Regs: registers;
begin
  with Regs do
    begin
      AH := $01;CH := $20;
    end;
  Intr($10,Regs);
end;

Procedure APPROX;
begin
  A := 0; Vstup := 0;
  for i := 7 downto 0 do
  begin
    Port[$378] := A + 1 shl i;
    delay(1);
    Vstup := Port[$379];
    if Vstup > 128 then A := A + 1 shl i;
  end;
end;

begin
  ClrScr;Zhasnikursor;
  writeln('Přerušení měření stiskem libovolné klávesy');
  repeat
    APPROX;gotoXY(1,3);
    writeln('napětí = ', round(RefNapeti *255/2.55 )/100 : 5:1,' V');
    until KeyPressed;
  Ch := ReadKey;
end.
```

obr. 33 až 36 jsou uvedeny programy pro řízení převodníků z obr. 6 a 9 a programy pro nastavení převodníků A/D z obr. 6 a 9 (BASIC, PASCAL).

Na základě dlouhodobého testu jednotek, které (u příležitosti výstavy učebních pomůcek realizovaných pracovníky katedry fyziky) byly zapojeny nepřetržitě celé dny, vyplýnulo, že chladič z hliníkového plechu tloušťky 2 mm je vhodnější nahradit chladičem tloušťky raději 2,5 nebo 3 mm, aby se snížila teplota obvodu A2030. Šroubek pro uchycení integrovaného obvodu je pak vhodné použít zapuštěný ze strany chladiče s maticí na izolační průchode obvodem A2030. Při použití desky 3 mm je nutné použít na vnější straně chladiče nízkou matici M3 bez podložek.

Výběr z literatury

- [1] Martier, H.: Centronics-Port-nicht nur zum Drucken. Elektronik 5/1991.
- [2] Bijkérk, B.: Centronics – A/D – D/A – Wandler. Elektor 5/1990.
- [3] Technické vybavení osobních počítačů typu PC/XT.
- [4] '286 Courier I/O II Card-User's Manual.
- [5] Punčochář, J.; Grůza, D.: Převodník A/D řízený počítačem. AR-A č. 4/1991.
- [6] Sheingold, D. H.: Analog-digital conversion handbook ANALOG DEVICES.
- [7] Petřík, J.: Jednotky styku mikropočítače IQ 151 s prostředím. Informace č. 2 až 9/ 1989.
- [8] Petřík, J.; Vrbík, V.: Mikropočítač ve výuce fyziky na Pedagogické fakultě v Plzni. Matematika a fyzika ve škole 2/1990.
- [9] Petřík, J.: Speciální periferní zařízení školních mikropočítačů. Sborník PF v Plzni – Fyzika VII.
- [10] AUTOSWITCH ENHANCED GRAPHICS CARD WITH PRINTER PORT. Edition 1.0, září 1988.
- [11] NEVA-Messen, Steuern, Regeln mit computer. Firemní literatura 1991.
- [12] NTL-computer physics. Firemní literatura 1991.
- [13] PHYWE-Experimentieren mit CENT. Firemní literatura 1991.
- [14] LEYBOLD – Schulversuche naturwissenschaften (Physik/Chemie). Firemní literatura 1991.
- [15] UNILAB-catalogue. Firemní literatura 1991.
- [16] TESTBED TECHNOLOGY – working for school technology. Firemní literatura 1991.
- [17] PHYWE computerunterstütztes Messen und Experimentieren mit COMEX. Firemní literatura 1991.
- [18] CORNELSEN EXPERIMENTA. Firemní literatura 1991.

Dodatek

Na katedře fyziky Pedagogické fakulty Západočeské univerzity v Plzni byly již krátce po získání prvních mikropočítačů navrženy a realizovány osmibitové převodníky A/D a D/A řízené modulem STAPER, užívané v laboratorních měření ve druhém a třetím ročníku aprobací MF a F-Zt.

Jakmile se v maloobchodní síti objevil dvanáctibitový převodník D/A MDAC565 a registr postupných approximací MHB1504, byla navržena nová jednotka styku mikropo-

čítače IQ 151 s prostředím, jejíž podrobný popis a stavební návod je v [7].

Jednotka, jejíž schéma, deska s plošnými spoji a rozložení součástek je na obr. 1D, obsahuje tříkanálový dvanáctibitový převodník A/D a osmibitový výkonový převodník D/A.

Jednotka tvoří jeden zásuvný modul do počítače IQ 151 a obsahuje na desce s plošnými spoji všechny aktivní i pasivní součástky. Vzhledem k tomu, že nebyla možnost realizovat desku s plošnými spoji na pracovišti jako oboustrannou, byla deska navržena jako jednostranná, doplněná drálovými propojkami.

Návrhu byla věnována velká péče, především dokonalému galvanickému a prostorovému oddělení analogové a číslicové části. Analogová a číslicová zem jsou spojeny v jediném místě (u obvodu MAC111).

Rezistory vstupního děliče jsou blízko sebe, aby byla kompenzována teplotní závislost děličního poměru.

Vstupní a výstupní svorky jsou nešeny profilem tvaru U. Na výkonový zesilovač MDA2020 je elektricky izolovaně upevněn duralový hranol, který zaručuje chlazení obvodu bez zátěže a na něj je přišroubována chladič deska s potenciálem 0 V, sloužící kromě obvodu tepla i ke stínění celého zařízení. Rovněž spodní strana musí být stíněna vzhledem k malým zpracovávaným napětím.

Popis zapojení

Jednotka styku s prostředím je napájena ze zdrojů mikropočítače, který má všechna potřebná napájecí napětí ± 12 V a ± 5 V. Tato napětí jsou filtrována na desce jednotky elektrolytickými, tantalovými a keramickými kondenzátory. V číslicové části je deska osazena programovatelným vstupním-výstupním obvodem MHB8255. Výběr adresy bran obvodu zajišťuje šestice invertorů MH7404 a osmivstupové hradlo NAND MH7430. Jeden z invertorů je využit pro inverzi signálu INIT, provádějícího inicializaci obvodu 8255.

Adresové vstupy A0, A1 ovládají přímo volbu brány obvodu. Dále jsou na vstup obvodu 8255 přivedeny ze sběrnice počítače signály pro čtení z obvodu a pro zápis do obvodu.

Stavové řídící slovo, nastavující požadovaný směr toku dat obvodem 8255, musí být na začátku každého uživatelského programu zasláno příkazem OUT7, 131, kde 7 je adresa řídícího registru a 131 nastavuje režim nula a směr toku dat následujícím způsobem:

Brána A je naprogramována jako výstupní a ovládá osmibitový převodník D/A, MDAC08. Její adresa je 4 a požadované binární slovo na vstupu MDAC08 se nastavuje v jazyce BASIC příkazem OUT4, číslo 0 až 255. V zapojení je MDAC08 spojen s převodníkem proud-napětí, tvořeném výkonovým zesilovačem MDA2020 a při referenčním napětí 10 V, určeném obvodem MAC01 a s uvedenými hodnotami obvodových prvků odpovídá zadanému číslu nula nulové napětí na výstupní svorce V a číslu 255 výstupní napětí +10 V. Toto výstupní napětí lze tedy měnit po 255 skocích 39 mV.

Brána B a čtverce nižších bitů brány C jsou naprogramovány jako vstupní a slouží ke čtení informace z registru postupných approximací, MHB1504. Vyšší bity brány

C jsou naprogramovány jako výstupní a mají tyto funkce:

Bit C4 startuje převod approximačního registru změnou z úrovně H na L, pak se vrací na úroveň H.

Bit C5, C6, C7 ovládají adresové vstupy osmikanálového analogového multiplexeru MAC08. Např. kanál S7, který připojuje na příkazem vstupního děliče R_2, R_3 pro rozsah 100 V se volí příkazem jazyka BASIC OUT6, 208, který kromě volby kanálu nastavuje bit C4 na úroveň H. Následným příkazem OUT6, 192 zůstává sepnutý kanál S7, ale změnou bitu C4 na úroveň L se odstartuje převod approximačního převodníku A/D. Následným příkazem OUT6, 208 se opět nastaví bit C4 na úroveň H, kanál S7 zůstává sepnut a připraven pro další převod.

Vzhledem k tomu, že při čtení bitů C0 až C3 z brány C příkazem INP (6) se přečte stav bitů C4 až C7, musí se při vyhodnocení odečíst příslušná kombinace bitů C4 až C7. Např. celé dvanáctibitové číslo, získané převodem napětí kanálu S7, lze pak číst příkazy (INP/6/-2 08/*256+ INP/5). Stejným způsobem se vyhodnocují i další kanály.

Kanál S 1 – OUT6,16 : OUT6,0 : OUT6,16
(INP/6/-16/*256+ INP/5)

Kanál S 2 – OUT6,48 : OUT6,32 : OUT6,48
(INP/6/-48/*256+ INP/5)

Kanál S 3 – OUT6,80 : OUT6,64 : OUT6,80
(INP/6/-80/*256+ INP/5)

Kanál S 4 – OUT6,112 : OUT6,96
: OUT6,112
(INP/6/-112/*256+ INP/5)

Kanál S 5 – OUT6,144 : OUT6,128
: OUT6,144
(INP/6/-144/*256+ INP/5)

Kanál S 6 – OUT6,176 : OUT6/160
: OUT6,176
(INP/6/-176/*256+ INP/5)

Kanál S 7 – OUT6,208 : OUT6,192
: OUT6,208
(INP/6/-208/*256+ INP/5)

Kanál S 8 – OUT6,240 : OUT6,224
: OUT6,240
(INP/6/-240/*256+ INP/5)

Tato tabulka přehledně ukazuje příkazy k adresaci kanálů multiplexeru, spuštění převodu a vyhodnocení měření.

V analogové části zapojení, realizující převod A/D, je použit dvanáctibitový převodník D/A, MDAC565, s vestavěným zdrojem referenčního napětí. Základní rozsah převodníku byl zvolen $\pm 2,5$ V z důvodu, které jsou uvedeny v popisu ochranných obvodů. Spojení s registrem postupných approximací a komparátorem MAC111 je realizováno doporučeným způsobem.

Signál o počátku a ukončení převodu (STATUS) je připojen na přerušovací systém mikropočítače a bude využíván pouze při záznamu rychlých dějů do paměti počítače, kdy se jednotka bude ovládat ve strojovém kódu. Při normální funkci (např. jako univerzální měřicí přístroj, ovládaný programem v jazyce BASIC) je vždy převod ukončen mnohem dříve, než se čte z bran obvodu 8255. Vlastní převodník je vybaven doporučenými odporovými trimry pro nastavení bipolární nuly a referenčního proudu.

Vstupní obvody

Vstupní obvody zajišťují úpravu velikosti měřeného napětí, ochranu zařízení proti nesprávnému zacházení, obvody přepínání kanálů a obvod absolutní hodnoty při měření střídavého napětí.

Jednotka má tři vstupní svorky – kanály X, Y, Z, společnou zemní svorku G a výstupní svorku pro převodník D/A, V.

Kanál X je vybaven programovým řízením přepínání rozsahů ± 250 mV, $\pm 2,5$ V,

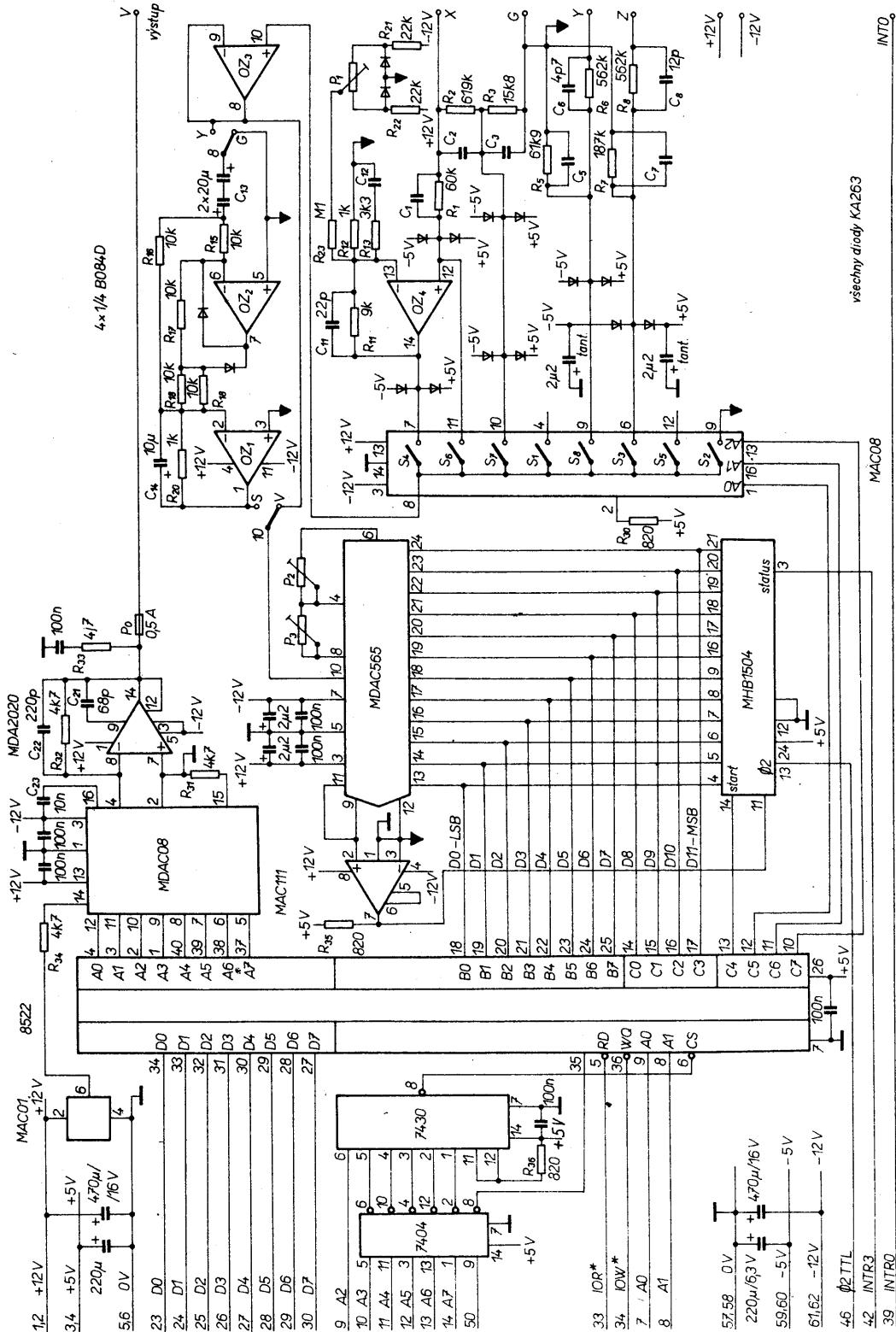
± 100 V (dovolené napětí pro školní účely), případně při měření střídavých napětí rozsahy 250 mV, 2,5 V, 100 V (st).

Vstupní signál ze svorky X je veden jednako na ochranný rezistor R_1 , na němž se vytváří příslušný úbytek napětí při překročení vstupního napětí asi ± 5 V, a jednako na dělič R_2 , R_3 , který slouží pro úpravu rozsahu 100 V.

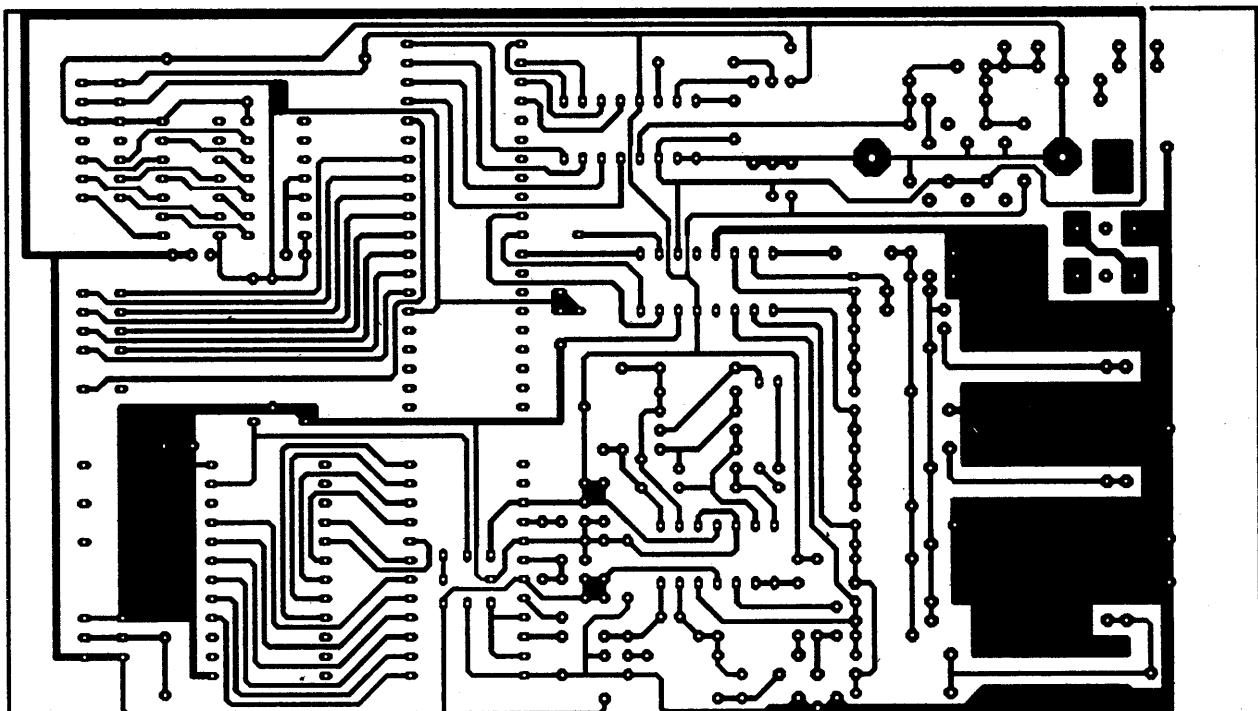
Signál o úrovni menší než 250 mV je zesílen v operačním zesilovači OZ₄ se vstupy osazenými polem řízenými tranzistory pro

získání velké vstupní impedance, v neinvertujícím zapojení se zesílením 10, určeném rezistory R_{11} a R_{12} . Rezistor R_{13} s kondenzátorem C_{12} slouží pro kompenzaci zesílení při měření střídavého napětí:

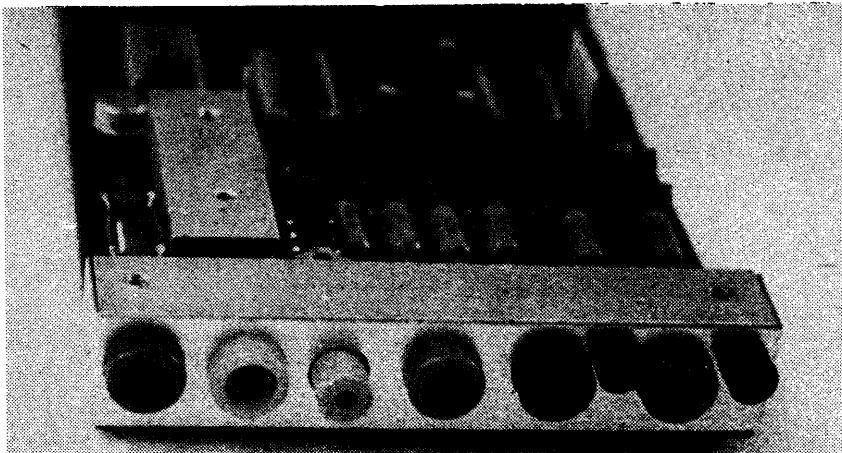
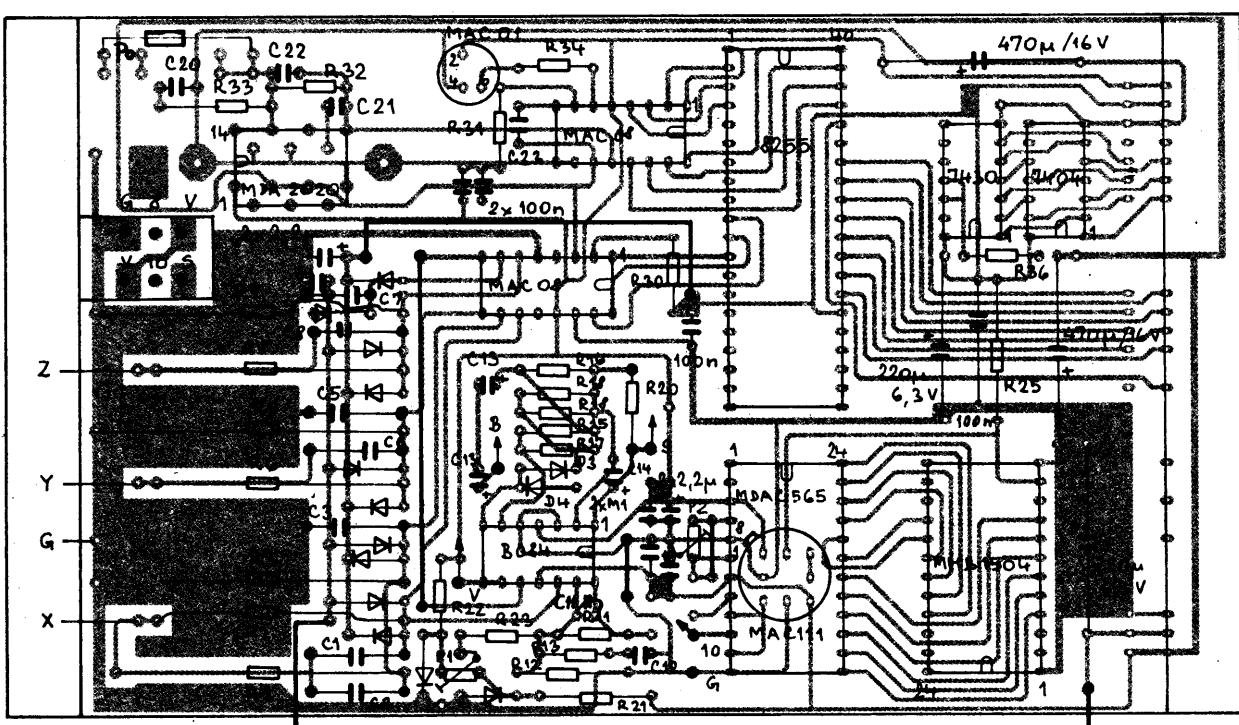
Poněvadž tento operační zesilovač je částečně kompenzován napěťovou nesymetrií, je možnost kompenzace zajistěna odpovědným trimrem P_1 , rezistory R_{21} , R_{22} , R_{23} a dvěma diodami.



Obr. 1D. Jednotka styku mikropočítače IQ151 s prostředím; a – schéma zapojení, b – deska s plošnými spoji (A201) a rozložení součástek



- 165



Vzhledem k tomu, že rezistory R_{11} a R_{12} určují přesnost zesílení a tím i přesnost měření na nejnižším rozsahu, byly použity destičkové rezistory typu WK68166 (oba rezistory na jedné podložce), $\pm 0,1\%$.

Kritická u těchto rezistorů je především teplotní závislost. Poněvadž jsou oba na jedné podložce, lze předpokládat, že budou mít v provozu i stejnou teplotu a stejný teplotní součinitel, takže napěťový přenos OZ₄ by měl být teplotně nezávislý. Tento předpoklad byl ověřen měřením děliče 9k/1k (WK68166) při oteplení na 60 °C. Měření

ukázalo, že teplotní závislost děličho poměru je více než o řád menší než teplotní závislost jednotlivých registrů. Aby přesto nebyla ovlivněna přesnost převodu vlastního převodníku MDAC565, je tento zesilovač užit pouze pro rozsah 250 mV a další rozsah, $\pm 2,5$ V, tj. základní rozsah převodníku, je připojen na kanál S 6 multiplexeru.

Na kanál S 7 je připojeno napětí z děliče R_2, R_3 , který zajišťuje rozsah 100 V. Vzhledem k napěťové bezpečnosti přístroje a k funkci rezistoru R_2 , který pracuje jako ochranný odpor pro kanál S 7 (a s ohledem na dostupnost) byly použity do vstupních děličů rezistory TR 163, 1 %.

Tyto rezistory ve vstupních děličích budou zmenšovat dosažitelnou přesnost měření jednak svojí absolutní hodnotou a tedy děličním poměrem (který může být nastaven pro určitý převodník a k němu odpovídající ovládací program cejchováním a dosazením programové konstanty), jednak teplotní závislosti. Proto byly použity rezistory s dostatečnou rezervou, aby se nezvyšovala jejich teplota procházejícím proudem. Rezistory jsou na desce s plošnými spoji co nejblíže u sebe, aby jejich teplota při umístění v počítači byla stejná a tím se zmenšila teplotní závislost děličního poměru. Kondenzátory C_1 až C_8 slouží ke kompenzaci napěťových děličů při měření střídavého napětí.

Signál z výstupu multiplexeru je veden na vstup sledovače, tvořeného dalším operačním zesilovačem se vstupem osazeným tranzistory řízenými polem, který zaručuje velkou vstupní impedanci.

Ze zapojení je zřejmé, že na rozsazích 250 mV a 2,5 V bude vstupní odpor mezi svorkou X a G určen prakticky sériovým spojením $R_2 + R_3$, tj. asi 600 k Ω .

Nejmenší vstupní odpor bude na rozsahu 100 V, kdy při překročení vstupního napětí ± 5 V se začne uplatňovat odpor rezistoru R_1 . Na tomto rozsahu bude tedy vstupní odpor určen paralelní kombinací R_1 a $R_2 + R_3$ (tj. bude přibližně 60 k Ω). Zvětšit vstupní odpor na tomto rozsahu by bylo možné zvětšením odporu rezistoru R_1 , ale jak bude uvedeno dále, to by vedlo ke zvětšení chyby měření vlivem závěrného proudu ochranných diod a vlivem zbytkových proudu multiplexeru.

Napětí z výstupu OZ₃ je při ss měření vedené přímo na vstup 10 obvodu MDAC565, nebo při měření střídavého napětí přes obvod absolutní hodnoty osazený OZ₁, OZ₂. Popis funkce tohoto obvodu i jeho nedostatky byly již v odborné literatuře popsány. Přesnost měření střídavých napětí je menší oproti měření stejnosměrných napětí. Proto také kompenzační kondenzátory ve vstupních děličích jsou pevně bez doladovacích trimrů. Předpokládá se měření harmonických napětí v rozsahu akustických kmitočtů.

Ochranné obvody

Ochranné obvody zajišťují, aby se na vstupu nesepnutého kanálu multiplexeru nemohlo objevit tak velké napětí, které by způsobilo jeho sepnutí.

Překročí-li při napájecím napětím ± 15 V napětí na vstupu nesepnutého kanálu mul-

tiplexeru 11 V, může být kanál sepnut. Při napájecím napětí ± 12 V se tato úroveň posune asi na 8 V.

Ochranné obvody rovněž zajišťují, aby napětí na vstupu zesilovače OZ₄ nepřesáhlo dovolenou mez.

Ochranné obvody však nesmí omezovat střídavé napětí při měření na střídavém rozsahu.

Vzhledem k tomu, že v počítači a tedy i na desce s plošnými spoji je k dispozici napětí ± 5 V, jsou ochranné obvody tvořeny diodami KA263 s velmi malým proudem v závěrném směru a příslušným ochranným rezistorem. Překročí-li napětí v místě spojení anody a katody ochranných diod ± 5 V, začne procházet proud diodou, která je v propustném směru, tento proud vytvoří na ochranném rezistoru úbytek napětí a v místě spojení anody a katody bude napětí maximálně (přibližně) $\pm 5,7$ V.

Proto byl zvolen základní rozsah převodníku MDA565 $\pm 2,5$ V, aby byl bezpečně zachován malý zbytkový proud diod v závěrném směru pro daný rozkmit stejnoměrného i střídavého napětí.

Ochranné obvody mohou způsobit vlivem konečné velikosti proudu diodami v závěrném směru přídavnou chybu měření. Je třeba volbou zvláště ochranného rezistoru zajistit, aby tato přídavná chyba (pokud možno) nezmenšovala přenos měření danou vlastnostmi vlastního převodníku.

Ochranné obvody by tedy neměly zmenšit přesnost měření o $\pm 1/2$ nejnižšího bitu.

Je zřejmé z předchozího popisu prvků zapojení, že tato přesnost bude zmenšena jak rezistory ve zpětné věti zesilovače na OZ₄, tak i rezistory vstupních děličů.

Koncepce zapojení byla tedy volena tak, aby alespoň na základním rozsahu převod-

níku $\pm 2,5$ V nebyla dosažitelná přesnost značně zmenšena.

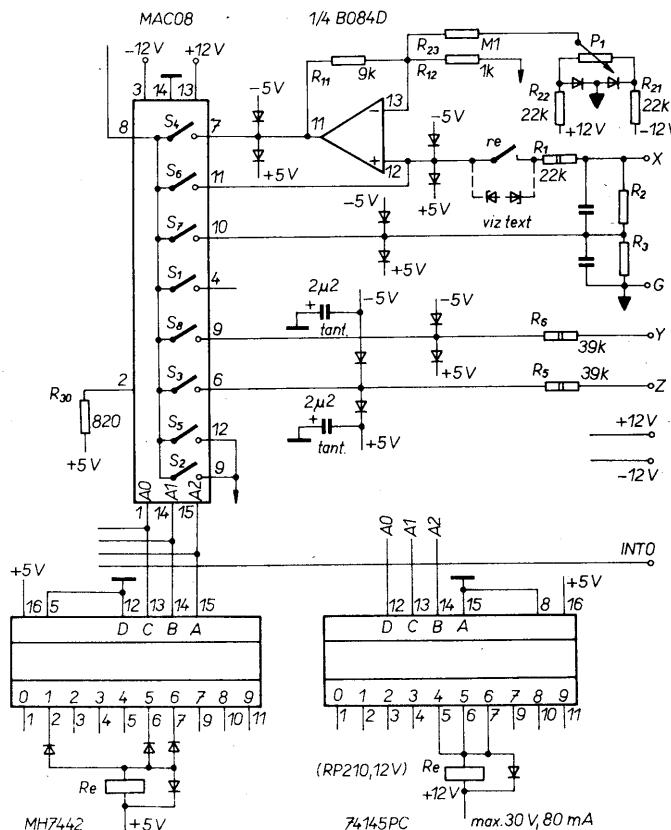
Podrobný popis funkce ochranných obvodů a jejich návrh z hlediska chyby způsobené svodovými proudy multiplexeru a proudy ochranných diod v závěrném směru je v lit. [7].

Na základě dlouhodobé práce s touto jednotkou byly upraveny vstupní obvody podle obr. 2D, týkající se hlavně ponechání rozsahu v kanálech Y a Z na 2,5 V a úpravy rozsahu v kanálu X na 65 V.

Důvody těchto změn byly popsány v lit. [7].

Programové vybavení pro tuto jednotku je uvedeno v úvodu příspěvku a v lit. [7] je uveden řídicí program, umožňující využít jednotky jako tříkanálový stejnosměrný i střídavý voltmetr a program pro řízení demonstračního experimentu „Jednostupňový tranzistorový zesilovač“.

Pro uvedenou jednotku byly vytvořeny program ve strojovém kódu, umožňující sejmout 1024 vzorků napětí, který je na obr. 3D (kanál X, rozsah 2,5 V). 1024 vzorků poskytuje k zobrazení průběhu napětí v délce trvání 0,1 s při vzorkovacím kmitočtu asi 20 kHz. Tento program rovněž obsahuje test na určení konstanty K, umožňující vyloučit chybu vlivem posuvu bipolární nuly a posuvu napěťové nesymetrie použitých OZ a výpočet maximální hodnoty ze sejmoutých vzorků. Pro měření harmonických průběhů je pak možné získat efektivní hodnotu vydělením $\sqrt{2}$. Při vzorkovacím kmitočtu 20 kHz je však omezen kmitočet měřeného napětí na 10 kHz (Nyquistův kmitočet). Obvod absolutní hodnoty, použitý v jednotce, umožňuje měřit střídavá napětí s kmitočtem vyšším než 20 kHz a je obvodově i součástkově nenáročný. Hodnoty uložené v paměti počí-



Obr. 2D. Úpravy jednotky styku

Obr. 3D. Program, umožňující sejmout 1024 vzorků napětí

4000	21	FE	44	LXI H,44FE	adresa počátku tabulky do HL
4003	3E	B0		MVI A,B0	volba kanálu X(sepnout S6)
4005	D3	06		OUT 06	
4007	3E	A0		MVI A,A0	start převodu
4009	D3	06		OUT 06	
400B	3E	B0		MVI A,B0	příprava pro další převod
400D	D3	06		OUT 06	
400F	00			NOP	prodloužení čas. intervalu
4010	DB	05		IN 05	čtení nižšího bytu (brána B)
4012	77			MOV M,A	přesun do paměti na adresu v HL
4013	23			INX H	zvýšení adresy
4014	DB	06		IN 06	čtení vyššího bytu (brána C)
4016	E6	0F		ANI OF	vymaskování bitů C4 až C7
4018	77			MOV M,A	přesun do paměti
4019	23			INX H	zvýšení adresy
401A	3E	4C		MVI A,4C	test na konec tabulky
401C	BC			CMP H	
401D	D2	07	40	JNC 4007	není konec, nový převod
4020	00			NOP	když je konec, určení K
4021	00			NOP	
4022	21	00	4D	LXI H,4D00	adresa uložení K
4025	3E	90		MVI A,90	volba kanálu S5
4027	D3	06		OUT 06	
4029	3E	80		MVI A,80	start převodu
402B	D3	06		OUT 06	
402D	3E	90		MVI A,90	příprava pro další převod
402F	D3	06		OUT 06	
4031	DB	05		IN 05	čtení nižšího bytu
4033	77			MOV M,A	uložení do paměti
4034	23			INX H	zvýšení adresy
4035	DB	06		IN 06	čtení vyššího bytu
4037	E6	0F		ANI OF	vymaskování bitů C4 až C7
4039	77			MOV M,A	uložení do paměti
403A	C3	40	40	JMP 4040	skok na výpočet max. hodnoty nebo RET návrat do BASIC
403D	00			NOP	
403E	FF			RST 7	
403F	0C			INR C	URČENÍ MAX. HODNOTY
4040	21	00	45	LXI H,4500	počátek tabulky do HL
4043	5E			MOV E,M	přesun nižšího bytu do E
4044	23			INX H	zvýšení adresy
4045	56			MOV D,M	přesun vyššího bytu do D
4046	23			INX H	zvýšení adresy
4047	4E			MOV C,M	přesun dalšího nižšího bytu do C
4048	23			INX H	zvýšení adresy
4049	46			MOV B,M	přesun dalšího vyššího bytu do B
404A	23			INX H	zvýšení adresy
404B	3E	4C		MVI A,4C	
404D	BC			CMP H	test na konec tabulky
404E	D2	58	40	JNC 4058	není konec, skok na 4058
4051	21	02	4D	LXI H,4D02	adresa uložení max. hodnoty do HL
4054	73			MOV M,E	přesun nižšího bytu na adresu v HL
4055	23			INX H	zvýšení adresy
4056	72			MOV M,D	přesun vyššího bytu max. hodnoty
4057	C9			RET	návrat
4058	7B			MOV A,E	výpočet většího z čísel v BC a DE
4059	91			SUB C	odečtení nižších bytů
405A	7A			MOV A,D	
405B	98			SBB B	odečtení vyšších bytů s přenosem
405C	D2	47	40	JNC 4047	není-li BC větší než DE další vzorky
405F	59			MOV E,C	jinak přesun BC do DE a nové vzorky
4060	50			MOV D,B	
4061	C3	47	40	JMP 4047	

tače je možné graficky zobrazit a zařízení použít jako digitální paměťový osciloskop, jak bude uvedeno dále.

K měření kmitočtu bylo využito magnetonového vstupu MGF1 počítače, a tím i vestavěného tvarovače. Program na obr. 4D umožňuje měřit signály o kmitočtech do 13 kHz bez jakéhokoliv přídavného zařízení (včetně zobrazení údaje na obrazovku). Úroveň přiváděného napětí musí být vzhledem k vestavěnému filtru v tvarovacím obvodu počítače 2 až 10 V pro signály vyšších kmitočtů, do asi 2 kHz stačí napětí řádu stovek mV.

Protože jednotka má vyvedena napájecí napětí ± 12 V, lze k ní připojit další přímo zasouvatelné moduly, které rozšiřují její využitelnost. Jedním z těchto modulů je programovatelný generátor harmonického napětí, řízený převodníkem D/A.

Programovatelné generátory střídavých napětí

Programovatelný generátor je zařízení, které umožňuje programem nastavit kmitočet napětí obvykle harmonického průběhu, případně i amplitudu tohoto napětí.

V podstatě libovolný průběh časově proměnného napětí lze generovat pomocí výkonového převodníku D/A v jednotce styku mikropočítače s prostředím. Do paměti počítače uložené okamžité hodnoty pro příslušné časové okamžiky (např. programem na obr. 5D) jsou pak dalším programem ve strojovém kódě na obr. 5D vybírány z paměti a zasílány na vstup převodníku D/A. Taktto získaný schodový průběh lze při generování harmonických průběhů vyfiltrovat a při kmitočtu hodinových impulsů mikropočítače 1 MHz dosáhnout kmitočtu asi 3,5 kHz.

Stejnosměrnou složku lze oddělit vazebním kondenzátorem.

Nevýhodou uvedeného způsobu je to, že procesor je vázán zvláště při vyšších kmitočtech na generování průběhu a nemůže být proto využit pro další výpočetní a grafické funkce. Proto byly navrženy a realizovány generátory pracující na principu převodníků napětí – kmitočet, které jsou řízeny programově měnitelným napětím z převodníku D/A v jednotce.

V zahraničí jsou pro tento účel vyráběny speciální IO (např. obvod 8038 – cena asi 15 DM). Pro jejich dosti vysokou cenu byly navrženy a realizovány generátory z běžně dostupných součástek s parametry srovnatelnými s obvodem 8038 a nižší cenou. Např. generátor uvedený v tomto příspěvku umožňuje přeladit celé akustické pásmo v jednom rozsahu a v asi 220 krocích, což obvodem 8038 jednoduše nelze.

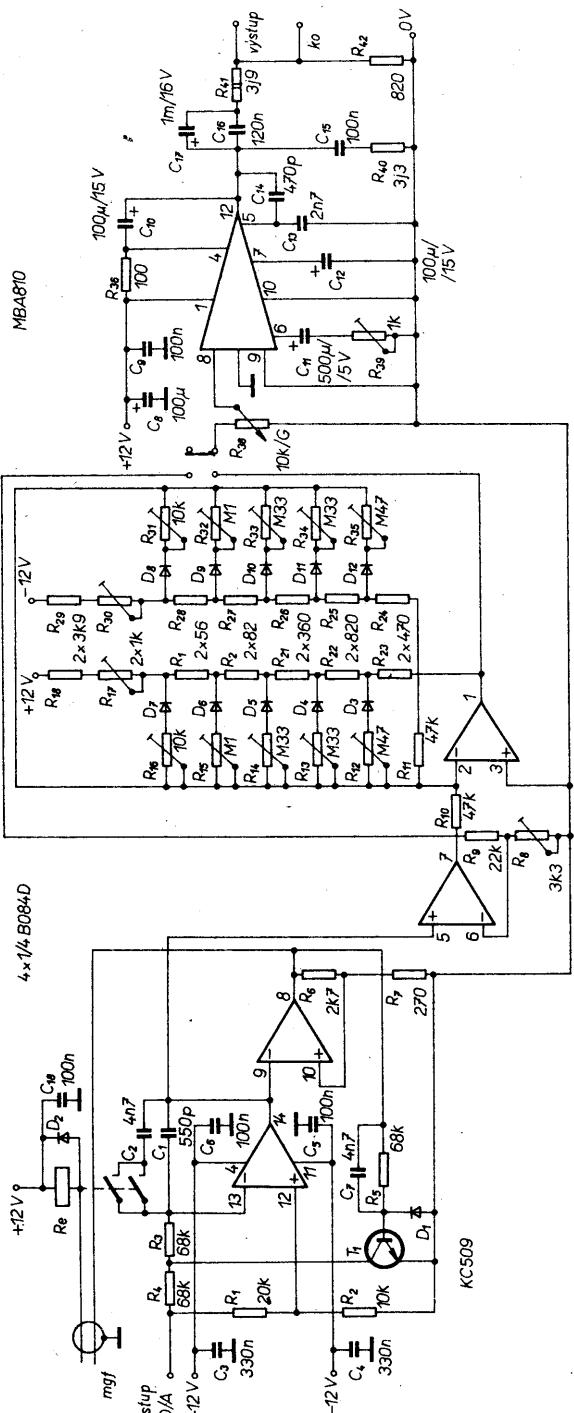
První funkční vzorek byl realizován poslučákou aprobace M-F Hanou Krůsovou a jeho schéma, deska s plošnými spoji a rozložení součástek je na obr. 6D. Tvoří jej obvyklé zapojení převodníku napětí-kmitočet (IO_1, IO_2), vytvářející na výstupu IO_1 napětí trojúhelníkovitého průběhu, které je po úpravě amplitudy v IO_3 přivedeno na převodník trojúhelník-sinusovka (IO_4). Regulátor výstupní amplitudy P_3 má logaritmický průběh, aby bylo možno lépe nastavit malé amplitudy výstupního napětí. Následu-

Obr. 4D. Program, umožňující měřit signály do 13 kHz

```

4000 21 00 00 LXI H,0000
4003 22 00 00 SHLD 0008 nulování HL
4006 JE FA MVI A,FA přepnutí selektoru dat "4157
4008 D3 86 OUT 86
400A DB 86 IN 86 čtení orány C vnitřního obvodu 8255
400C EE 56 XRI 56 vyhodnocení stavu bitu C7
400E F2 0A 40 JP 400A je-li c7 0, nové čtení
4011 23 INX H je-li c7 1, zvýšení HL o 1
4012 3A 00 00 LDA 0008 Vyhodnocení požadala času
4015 FE 19 CPI 19 není 1 s, skok na 401F
4017 DA 1F 40 JC 401F je 1 s, přepnutí selektoru dat
401A JE F8 MVI A,F8
401C D3 86 OUT 86
401E C9 RET návrat do BASIC
401F DB 86 IN 86 čtení brány C
4021 EE D6 XRI D6 vyhodnocení stavu bitu C7
4023 F2 1F 40 JP 401F je-li c7 1, nové čtení
4026 C3 0A 40 JMP 400A je-li c7 0, skok na 400A

```



Obr. 5D.

```

0 CLS. OUT7,131
10 FOR I=0 TO 36. X=127+INT(127*SIN((PI/180)*10*I))
20 POKE(19712+I),X:NEXT X
30 CALLHEX(4C00)

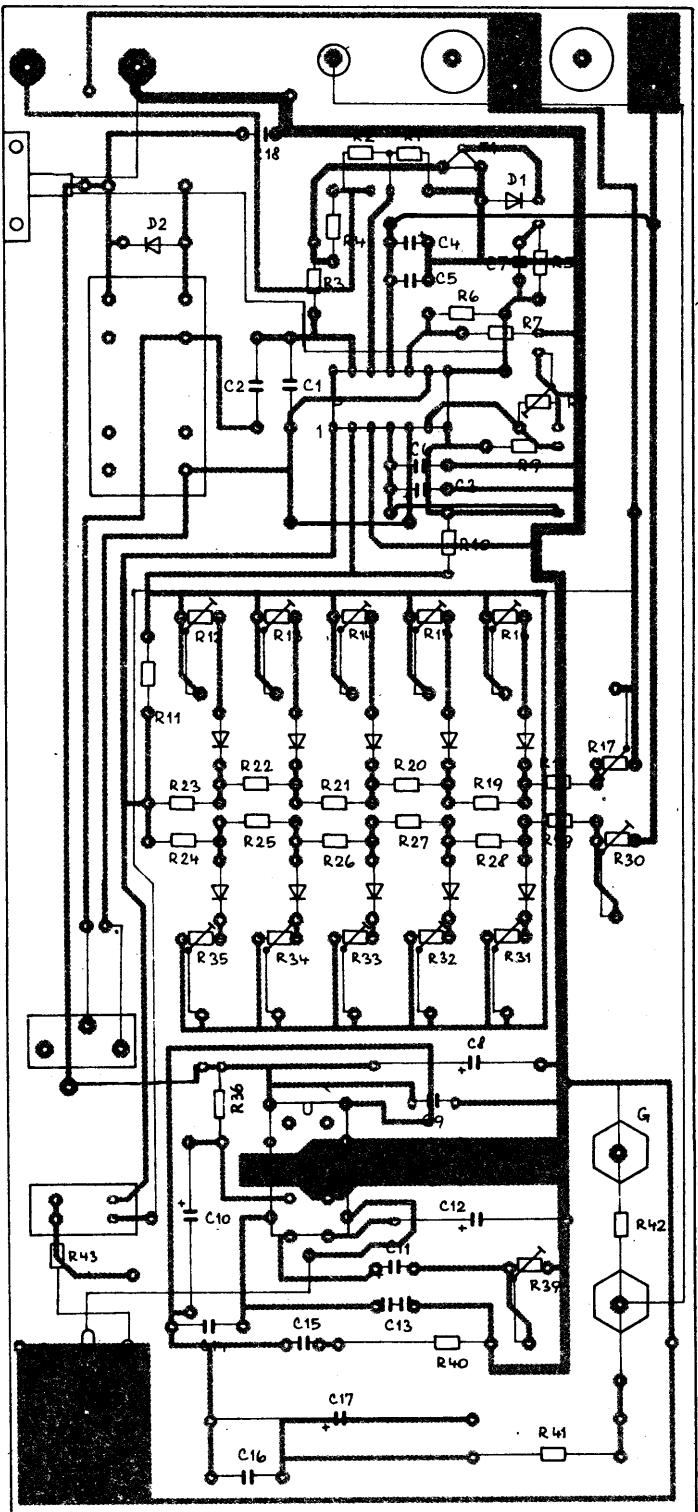
```

```

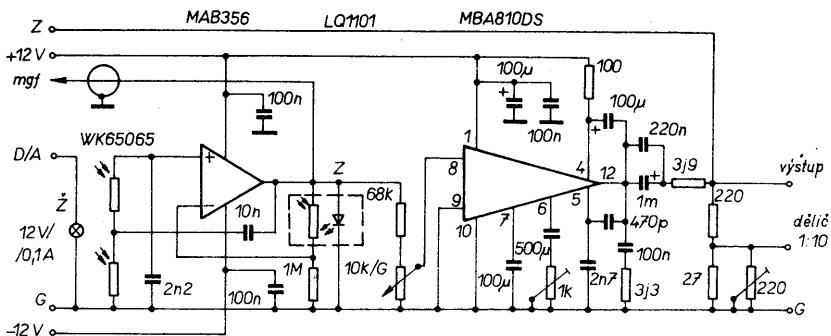
4C00 01 24 4D LXI B,4D24
4C03 0A LDAX B
4C04 D3 04 OUT 04
4C06 0D DCR C
4C07 C2 03 4C JNZ 4C03
4C0A C3 00 4C JMP 4C00

```

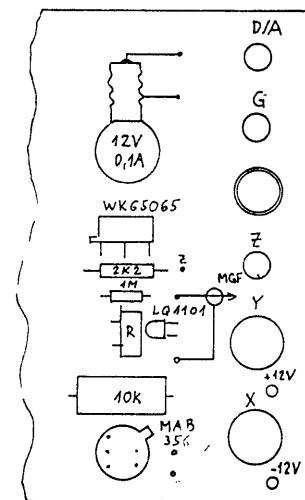
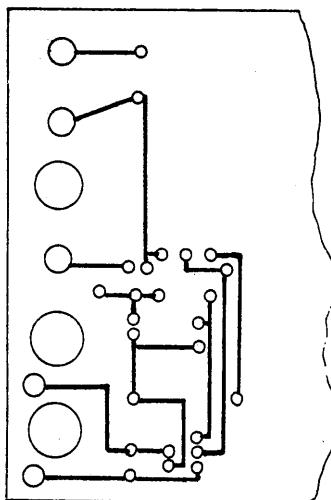
adresa počátku tabulky do BC
přesun čísla na adresu v BC do A
výstup do D/A převodníku
snížení adresy
test na konec tabulky
není konec, další číslo do A



Obr. 6D. Programovatelný generátor; a – schéma zapojení, b – deska s plošnými spoji



Obr. 7D. Generátor harmonického napětí



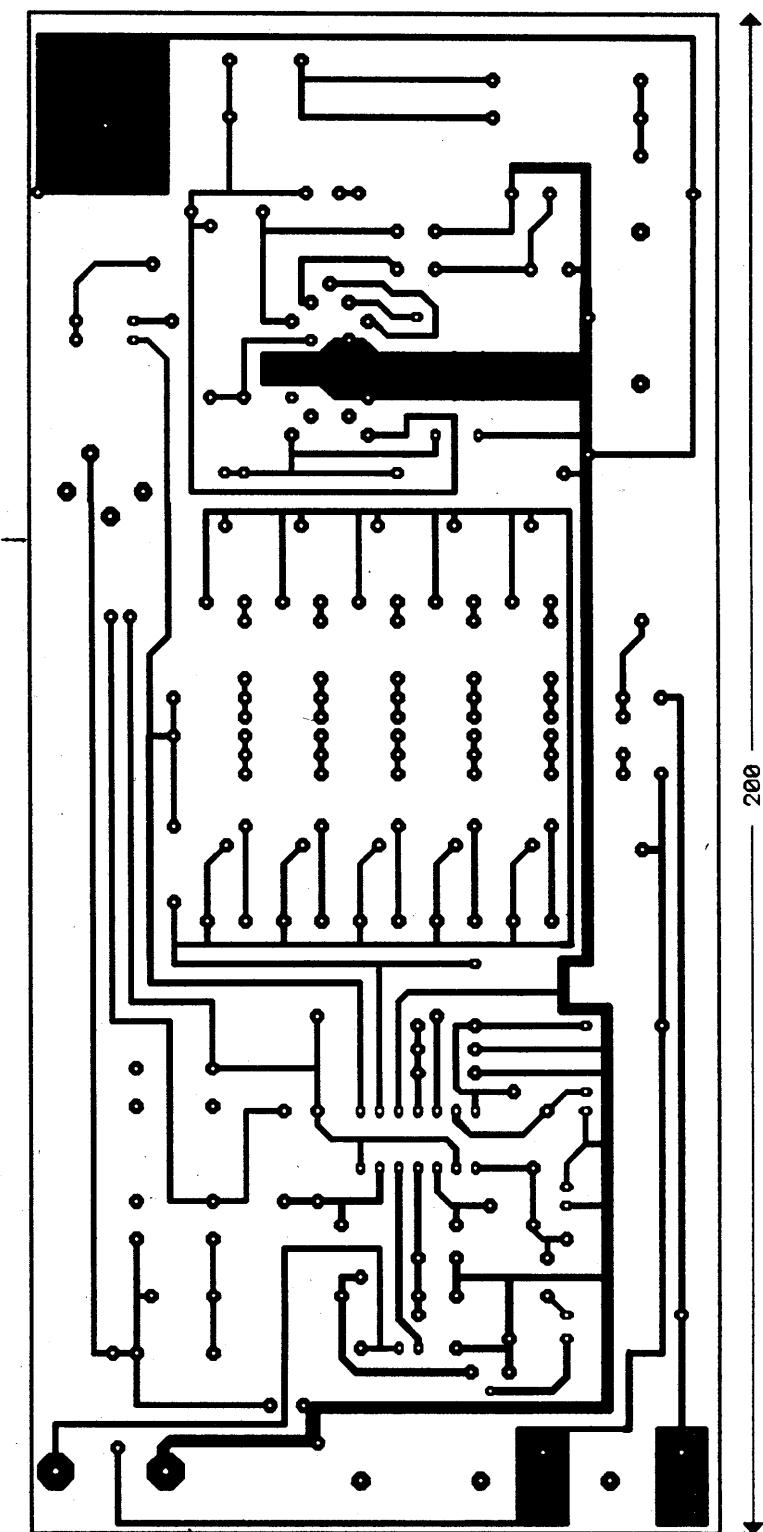
Obr. 8D. Úprava desky s plošnými spoji

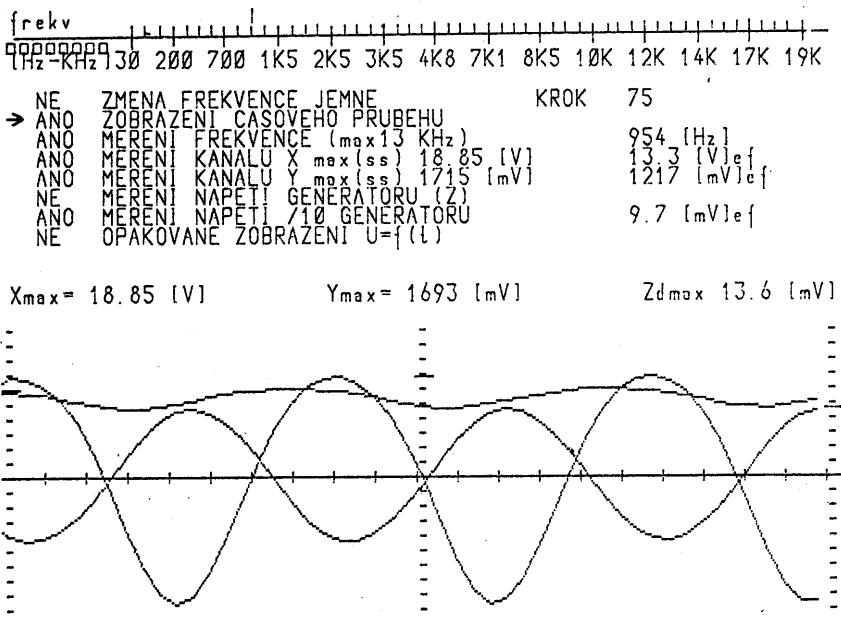
je impedanční převodník, tvořený integrovaným výkonovým zesilovačem MBA810 v doporučeném zapojení. Na jeho výstupu je signál s malým výstupním odporem, tvořeným prakticky pouze ochranným rezistorem R_{41} . Maximální amplituda je nastavena na 2,5 V trimrem R_{39} .

Impulzy z výstupu Schmittova klopného obvodu jsou přivedeny do magnetofonového vstupu počítače pro měření kmitočtu generovaného napětí.

Z parametru získaných měřením na tomto funkčním vzorku vyplynula neúměrná složitost zapojení, značný počet součástek i velká zástavěná plocha desky s plošnými spoji. Proto byl navržen a realizován nový funkční vzorek generátoru harmonického napěti, jehož schéma je na obr. 7D, u něhož je dosaženo zmíny kmitočtu změnou odporu dvojice běžných fotorezistorů WK65065 (2 fotorezistory v 1 pouzdře), zapojených v oscilátoru s přemostěným T článkem (viz Konstrukční příloha AR/1986, kmitočet měníme dvojitým potenciometrem).

Změna odporu fotorezistoru je způsobena změnou napětí na běžné žárovce 12 V/0,1 A, připojené k výstupu výkonového převodníku D/A v jednotce styku. Je přitom vhodné využít nonlinearitu změny odporu fotorezistoru na napětí na žárovce. S touto žárovkou bylo možné přeladit celé pásmo 20 Hz až 20 kHz v přibližně 220 krocích.





Obr. 9D. Zobrazení průběhů na obrazovce

Obr. 10. Výpis programu (BASIC) k využití jednotky styku jako automatického osciloskopu (i k řízení programovatelného generátoru)

Pro řízení tohoto generátoru převodníkem D/A z jednotky styku pro počítače PC AT/XT, popisovaným v tomto příspěvku, je možné přímo připojit žárovku 12 V na převodník D/A, ale pak se využije pouze asi 120 kroků pro přeladění celého akustického pásma. Vhodnější je použít místo žárovky 12 V žárovku 24 V a jedním pólem ji připojit na napětí – 15 V a druhým pólem na výstup převodníku D/A v jednotce. Pak se opět využije asi 250 kroků pro změnu kmitočtu od 20 H do 20 kHz.

Aby se zmenšil vliv teploty žárovky na změnu odporu fotorezistoru, je umístěna nad fotorezistory (obr. 8D.). Je připájena na tlustší dráty, aby bylo možno při sežizování generátoru nastavit její vzdálenost od fotorezistorů a tím i generovaný signál nejvyššího kmitočtu 20 kHz. Případně větší clouhodobé časové změny fotorezistorů jsou kompenzovány programovou regulační smyčkou.

Amplituda je opět stabilizována fotorezistorem, osvětleným světelnou diodou LQ1101. Tento „optočlen“ musí být uzavřen v dokonale nepřůhledném pouzdře.

Obvod regulace amplitudy a impedanční převodeník byly převzaty z prvního funkčního vzorku.

Konstrukčně je generátor proveden jako přímozasouvatelný malý modul do jednotky styku. Jeho zasunutím se přivedou na generátor potřebná napájecí napětí, řídící napětí z převodníku D/A a výstupní (generované) napětí se přivede na kanál Z.

Programové vybavení

Pro řízení programovatelného generátoru byl vypracován ovládací program, který umožnuje využít školního mikropočítače nejen k programové změně kmitočtu, ale také jako:

1. paměťového tříkanálového digitálního osciloskopu s automatickým přepínáním rozsahů a s automatickou indikací překročení měřitelných parametrů, automatickou volbou měřítka na ose napětí v každém kanálu, s automatickou volbou časové základny;
2. digitálního měřiče kmitočtu;
3. digitálního stejnosměrného voltmetru s automatickým rozlišením polarity a s automatickým přepínáním rozsahů;
4. digitálního střídalového voltmetru s automatickým přepínáním rozsahů, automatickou indikací překročení parametrů, s digitálním zobrazením vrcholové hodnoty měřeného napětí a efektivní hodnoty harmonického průběhu, se zobrazením vrcholové hodnoty měřeného napětí do stupnice napětí při zobrazení časového průběhu;
5. průběhy na obrazovce je možné pomocí programu HARD COPY (KOMENIUM) vykreslit na grafické jednotce XY 4131 (obr. 9D);
6. měřící režim se volí kurzorovými tlačítky ↑ ↓, tlačítky A (ano) N (ne) a je indikován na obrazovce;
7. kmitočet se volí kurzorovými tlačítky → ← a je indikován polohou značky na stupnici kmitočtu. Volba může být hrubá pro rychlé přeladění nebo jemná pro přesné nastavení. Stupnice může být pouze orientační, poněvadž kmitočet generovaného signálu je možné digitálně ihned měřit;
8. podle zvoleného měřicího režimu je možné při opakování měření časového

```

680 IF B=4 THEN1290
720 IF B=5 THEN1490
760 IF B=6 THEN1590
800 IF B=7 THEN1390
810 IF B=8 THEN1690
820 IF B=9 THEN1790
850 RETURN
900 IFQ=0ANDR=0THENRETURN
910 IFQ=0ANDR=1THEN1020
920 IFQ=1ANDR=1THEN1000
930 RETURN
1000 IFV=0ANDX=0ANDY=0ANDZ=0ANDQ=1THENPRINT&5,46"ZVOLTE KANAL",CHR(7).RETURN
1010 IF Z=1 THEN1040
1012 IF Y=1 THEN1140
1013 IF V=1 THEN1100
1014 IF X=1 THEN1160
1015 RETURN
1020 PRINT&5,2"NE "&5,40" "&6,2"NE "
1021 PRINT&9,2"NE "&9,46" "
1022 PRINT&6,46" "
1023 PRINT&7,2"NE "&7,46" "&10,2"NE "&10,46"
1024 PRINT&8,2"NE "&8,46" ".X=0,Y=0,Z=0,F2=F1,V=0,U=0
1025 ERASE,PEN2, DRAW2*D,250, LABEL "T", Q=0, R=0, M=0
1026 PRINT&14,0"
1027 PRINT&7,23" "
1028 PRINT&8,23" "
1030 PRINT&29,1"VOLBA MERICIMO REZIMU - SVISLE SIPKY + TLACITKA A(ANO) N(NE)
1035 PRINT&28,1"VOLBA FREKVENCE - VODOROVNE SIPKY".RETURN
1040 PRINT&5,45" "&5,44
1045 PRINT&5,2"ANO"&9,2"ANO"&6,2"ANO":M=1:GOSUB11000:GOSUB6000
1046 IFY=1THENGOSUB6105
1047 IFV=1THENGOSUB6110
1048 IFX=1THENGOSUB6120
1050 Q=0,R=0,RETURN
1100 PRINT&10,45" "&10,44
1110 PRINT&5,2"ANO"&10,2"ANO"&6,2"ANO":M=1:GOSUB11000:GOSUB6000
1117 IFX=1THENGOSUB6120
1120 Q=0,R=0,RETURN
1140 PRINT&5,45" "&5,44
1145 PRINT&5,2"ANO"&8,2"ANO"&6,2"ANO":M=1:GOSUB11000:GOSUB6000
1146 IFV=1THENGOSUB6110
1147 IFX=1THENGOSUB6120
1150 Q=0,R=0,RETURN
1160 PRINT&5,45" "&5,44
1165 PRINT&5,2"ANO"&7,2"ANO"&6,2"ANO":M=1:GOSUB11000:GOSUB6000
1170 Q=0,R=0,RETURN
1190 IFR=0 THEN J=J:GOTO1230
1200 IF Q=1 THEN J=1:GOTO1270
1220 J=0
1230 IFJ=1 THEN 1270
1240 PRINT&4,2"NE ",G=5,RETURN
1270 PRINT&4,2"ANO",G=1,RETURN
1290 IF R=0 THEN M=M:GOTO 1330
1300 IFQ=1THEN M=1:GOTO 1370
1320 M=0
1330 IFM=1 THEN 1370
1340 PRINT&6,2"NE ".PRINT&6,46" ",RETURN
1370 PRINT&6,2"ANO":GOSUB11000,RETURN
1390 IFR=0THENZ=Z:GOTO1430
1400 IFQ=1THEN Z=1:GOTO1470
1420 Z=0
1430 IFZ=1 THEN1470
1440 PRINT&9,2"NE ".PRINT&9,46" ",RETURN
1470 PRINT&9,2"ANO".RETURN
1490 IFR=0THENX=X:GOTO1530
1500 IF Q=1 THEN X=1:GOTO1570
1520 X=0
1530 IFX=1 THEN1570
1540 PRINT&7,2"NE ".PRINT&7,46" ",RETURN
1550 PRINT&7,23" ",RETURN
1570 PRINT&7,2"ANO".RETURN
1590 IFR=0 THENY=Y:GOTO1630
1600 IFQ=1 THENY=1:GOTO1670
1620 Y=0
1630 IFY=1THEN1670
1640 PRINT&6,2"NE ".PRINT&6,46" ",RETURN
1650 PRINT&6,23" ",RETURN
1670 PRINT&6,2"ANO":Q=0,R=0,RETURN
1690 IFR=0 THENV=V:GOTO1730
1700 IFQ=1 THENV=1:GOTO1770
1720 V=0
1730 IFV=1THEN1770
1740 V=0,PRINT&10,2"NE ".PRINT&10,46" ",RETURN
1770 PRINT&10,2"ANO":Q=0,R=0,RETURN
1790 IFQ=0ANDR=0THENRETURN
1800 IFQ=0ANDR=1THENPRINT&11,2"NE ",U=0:GOTO1820
1810 IFQ=1ANDR=1THENPRINT&11,2"ANO",U=1,RETURN
1820 RETURN
6000 PRINT&29,1"
6010 PRINT&29,1"
6050 PRINT&5,46" "&5,45
6052 IFF1>10000THENPRINT&5,46"<MEZ ZOBRAZENI "&5,45:RETURN
6053 IFF1>15000THENPRINT&5,46"<MEZ KV.ZOBRAZENI":A1=20,A9=17762,A8=1:GOT06095
6055 IF F1>10000 AND F1<1500 THEN A1=10,A9=17762,A8=1:GOT06100
6060 IF F1>5000 AND F1<1000 THEN A1=5,A9=17762,A8=1:GOT06100
6065 IF F1>3000 AND F1<500 THEN A1=3,A9=17862,A8=1:GOT06100
6070 IF F1>1500 AND F1<300 THEN A1=1.5,A9=18062,A8=1:GOT06100
6075 IF F1>1000 AND F1<150 THEN A1=.9,A9=18262,A8=2:GOT06100
6080 IF F1>700 AND F1<100 THEN A1=.7,A9=18662,A8=3:GOT06100
6085 IF F1>300 AND F1<700 THEN A1=.3,A9=19410,A8=5:GOT06100
6090 A1=.25,A9=19710,A8=6
6095 PRINT&5,45
6100 IFV=1THEN A5=500,A6=53,PRINT&14,48"Zmax=".GOSUB20000:GOT06135
6105 IFY=1THEN A5=250,A6=29,PRINT&14,24"Ymax=".GOSUB20150:GOT06135
6110 IFV=1THEN A5=500,A6=53,PRINT&14,48"Zmax".GOSUB20200,A7=.1:GOT06135
6120 IFX=1THEN A5=0,A6=5,PRINT&14,14,0"Xmax=".GOSUB20050:GOT06135
6135 IFW>1000THENZ3=.03:GOT06180
6140 IFABS(W)>440THENZ3=.06125:GOT06180
6145 IFABS(W)>220THENZ3=.125:GOT06180
6150 IFABS(W)>110THENZ3=.25:GOT06180
6160 IFABS(W)>40 THENZ3=.5:GOT06180
6170 Z3=1

```

průběhu zobrazit maximálně čtyři průběhy do jednoho obrázku na obrazovce;

9. při zvoleném režimu pouze časového zobrazení je možné do jednoho obrázku zobrazit libovolný počet průběhů. Například při volbě jednoho až tří kanálů při časovém zobrazení se automaticky bez zásahu obsluhy změří a vytiskne na obrazovku kmitočet generovaného napětí, automaticky se vypočte a nastaví vhodná časová základna, změří a vytiskne maximální amplituda zobrazovaného průběhu, vyznačí se na ose napětí příslušného kanálu, vypočte se efektivní hodnota a rovněž se vytiskne na obrazovku, zvolí se opět automaticky nejvhodnější měřítka na ose napětí a vykreslí se na obrazovce změřený průběh. Totéž probíhá při měření v každém kanálu, takže každý zobrazený průběh má svoji stupnice napětí s vyznačenou vrcholovou hodnotou.

Dosud určovali studenti průběhy napětí pouze na obrazovace osciloskopu, která nedovolovala jednak zobrazit tři kanály, jednak (vzhledem k velikosti) nebylo možné rozuměně rozbrazit stejnosměrné i střídavé složky. K měření byly dále nutné další přístroje, ss voltmetr pro změření pracovního bodu, střídavý voltmetr pro měření vstupního pracovního bodu, střídavý voltmetr pro měření vstupního napětí a rozkmit výstupního napětí se určoval značně nepřesně z obrazovky osciloskopu.

Je zřejmé, že nový způsob měření zvětšuje jak přesnost měření a názornost experimentu, tak didaktickou účinnost prováděních měření.

V tomto programu bylo využito převodníku D/A v jednotce k programovému řízení kmitočtu, jednoho kanálu převodníku A/D k digitálnímu měření a zobrazení časového průběhu výstupního generovaného napětí a dalších dvou kanálů bylo využito k měření a zobrazení časových průběhů na měřeném objektu. Magnetofonového vstupu mikropočítače bylo opět využito k digitálnímu měření kmitočtu.

Programy ve strojovém kódu pro sejmoutí vzorků, výpočet amplitud a měření kmitočtu se nahrávají současně s programem v jazyce BASIC (řádky 10 až 75).

Celou část určenou pro zpracování analogového signálu, tj. vstupní obvody, multiplexer, obvod absolutní hodnoty, impedanční převodník, vstupní zesilovač, vlastní převodník A/D i převodník D/A s výkonovým zesilovačem je možné připojit i k jiným počítačům, pokud jsou vybaveny paralelním rozhraním, tvořeným obvodem MHB8255. Je možné využít i celého ovládacího programu, změnění

```

6180 K2=INT(K1*.0317).DRAW511,K2,-2,DRAW0,K2,-1
6185 DRAW0+A5,(Z1-K1)*Z3+K2,-2
6187 IFAT=26THEN6195
6190 DRAW11+A5,(Z1-K1)*Z3+K2,-1:PRINT&14,A6INT(W#1.2207)*A7["mV"]
6192 GOTO6200
6195 DRAW11+A5,(Z1-K1)*Z3+K2,-1
6197 PRINT&14,A6INT((W#1.318)*2+.5)/2/10"["V"]
6200 FORI=K2-60 TO K2+60 STEP 6: DRAW3+A5,I,-2,DRAW6+A5,I,-1:NEXTI
6210 FORI=24TO50STEP25:DRAW1,K2-2,LABEL"+":NEXTI
6215 DRAW0,K2,-2
6220 FORI=17662TO A9 STEP (2*AB)
6230 Z2=(PEEK(I)+PEEK(I+1)*256-K1)*23
6240 DRAW(I-17663)*A1,Z2+K2,-1
6250 NEXTI
6300 RETURN
10000 F1=2WORD(HEX(4000))
10005 IFF1>13200THENPRINT&6,46"< MEZ MERITELNOSTI"&6,45:RETURN
10010 PRINT&6,46F1["Hz"] ":"PRINT&6,45:RETURN
11000 IFM=0THEN RETURN
11010 WAIT(10)
11015 F2=2WORD(HEX(4000))
11020 IFF2>13200THENPRINT&6,46"< MEZ MERITELNOSTI"&6,45:RETURN
11030 F1=F2:PRINT&6,46F1["Hz"] ":"PRINT&6,45:RETURN
12000 F2=2WORD(HEX(4000))
12010 IFF2>13200THENPRINT&6,46"< MEZ MERITELNOSTI"&6,45:RETURN
12020 F1=F2:RETURN
12030 F1=F2:RETURN
20000 OUT6,80,POKE16457,80,POKE16461,64,POKE16465,80
20005 POKE16491,48,POKE16495,32,POKE16499,48,COSUB20500
20010 IFZ1>4094THENPRINT&9,46"PREKROCEN ROZSAH":W=2048,A7=1:RETURN
20020 PRINT&9,46INT(W#866552) ["mV]af "&9,45,A7=1:RETURN
20050 OUT6,112,POKE16457,112,POKE16461,96,POKE16465,112
20055 POKE16491,144,POKE16495,128,POKE16499,144,COSUB20500
20060 IFZ1>4094THENOUT6,176,POKE16457,176,POKE16461,160,POKE16465,176,GOTO20086
20065 PRINT&7,23"max(ss)"(INT(W#1.2207)+.5)/10 ["mV"]
20070 PRINT&7,46(INT(W#866552)+.5)/10 ["mV]af "&7,45,A7=1:RETURN
20080 POKE16491,144,POKE16495,128,POKE16499,144,COSUB20500
20090 IFZ1>4094THENOUT6,208,POKE16457,208,POKE16461,192,POKE16465,208,GOTO20126
20095 IF ABS(W)<240 THEN 20050
20100 PRINT&7,23"max(ss)"(INT(W#1.2207)) ["mV"]
20110 PRINT&7,46(INT(W#866552)) ["mV]af "&7,45,A7=1:RETURN
20120 POKE16491,48,POKE16495,32,POKE16499,48,COSUB20500
20130 IF Z1>4094THENPRINT&7,46"PREKROCEN ROZSAH":RETURN
20132 IFABS(W)>0THENOUT6,176,GOTO20550
20135 PRINT&7,23"max(ss)"(INT(W#1.318)*2+.5)/2/10"["V"]
20140 PRINT&7,46(INT(W#22486)*2+.5)/2/10"["V]af "&7,45,A7=26:RETURN
20150 OUT6,240,POKE16457,240,POKE16461,224,POKE16465,240
20155 POKE16491,48,POKE16495,32,POKE16499,48,COSUB20500
20160 IFZ1>4094THENPRINT&8,46"PREKROCEN ROZSAH":RETURN
20170 PRINT&8,23"max(ss)"INT(W#1.2207) ["mV]
20180 PRINT&8,46INT(W#866552) ["mV]af "&8,45,A7=1:RETURN
20200 OUT6,80,POKE16457,80,POKE16461,64,POKE16465,80
20210 POKE16491,48,POKE16495,32,POKE16499,48,COSUB20500
20220 IFZ1>4094THENPRINT&10,46"PREKROCEN ROZSAH":W=2048,A7=1:RETURN
20230 PRINT&10,46INT(W#866552)/10 ["mV]af "&10,45,A7=1:RETURN
20250 CALLHEX(4045)
20510 K1=PEEK(HEX(4D00))+PEEK(HEX(4D01))*256
20520 Z1=PEEK(HEX(4D02))+PEEK(HEX(4D03))*256
20530 W=Z1-K1:RETURN
20550 POKE16457,176,POKE16461,160,POKE16465,176,GOTO20080

```

UPOZORNĚNÍ

Destičku E200 s plošnými spoji (40,-), stavebnici převodníku (450,-), základní programové vybavení (v rozsahu, uveřejněném v tomto čísle) (200,-), programy pro využití převodníku jako automatickou st nebo ss osciloskopu (200,-), programový komplet (350,-), osazenou a oživenou desku AD/DA převodníku + základní programy (995,-), s můžete objednat na adresu firmy EMGO, Inovační centrum VÚHŽ, 739 51 Dobrá, tel. (0658) 23421/kl.340 nebo fax (0658)23016. Programové vybavení dodáváme na disketu 5,25" nebo 3,5". Na požadání zašleme i další informace o dodávkách modulů, které po jednoduchém připojení zasunutím do zdířek převodníku rozšíří jeho využití, například ve školní výuce. Namátkou vybíráme nf generátor, vf generátor, několik variant teploměrů.

Všechny další informace můžete dostat na adresu ing. Josef Petřík, Zborovská 94, 301 34 Plzeň.

Autor navíc upozorňuje, že v popisované jednotce nejsou vestavěny žádné filtry ani

zpožďovací obvody (z důvodů použití jako digitální osciloskop). Proto se musí střídavá i stejnosměrná napětí (především malých velikostí) přivádět stíněným vodičem, jinak by mohly být údaje, indikované na displeji, ovlivněny výskytem rušivých polí.

Běžné střídavé a především stejnosměrné voltmetry mají takové filtry vestavěny a střídavé voltmetry bývají vybaveny obvody pro získání absolutní hodnoty s velkou časovou konstantou, eliminující krátkodobá rušivá napětí. Takový digitální filtr je použit ve výše uvedeném programu XYZRG1ST, pracuje však pouze pro stejnosměrná napětí.

Stejně tak není použit žádný filtr pro signální vyšší kmitočtů než je Nyquistův kmitočet, protože jeho velikost závisí na hodinovém kmitočtu počítače a byl by tedy pro různé počítače různý. Není použit také žádný vazební kondenzátor a program tedy zajišťuje měření i stejnosměrné složky.

Pro měření pouze střídavých složek napětí je vhodné zhotovit např. měřicí stíněný kabel, který má vestavěn kondenzátor o kapacitě asi 100 pF pro oddělení stejnosměrné složky.

K AR B1-1992

K prvnímu číslu AR řady B jsme dostali do redakce několik stran připomínek od našeho čtenáře Jindřicha Wernitzera z Jablonce n. N. Po konzultaci s lektory a autorem vyjímáme z jeho připomínek:

li se adresy příslušné jednotlivým branám a syntaxe některých příkazů (zvláštně grafických).

K počítači ZX Spectrum lze připojit jednotku pomocí interface s MHB8255 podle AR-A č. 6/1985 nebo v Příloze AR Mikroelektronika 1988, změní-li se adresa řídicího registru CWR 127 dec, tj. pro naprogramování obvodu se zašle řídicí slovo příkazem OUT 127, 131. Brána A má adresu 31 dec., B má adresu 63 dec. a C má adresu 95 dec.

Čtení z bran B a C např. pro volbu kanálu X rozsah 2,5 V včetně testu na K má pak tvar:

```

OUT 95,144 : OUT 95,128 : OUT
95,144 : LET K = /(IN 95 - 144) * 256 + IN
63 : OUT 95,175 : OUT 95,160 : OUT
95,176 : LET X = /(IN 95 - 175) * 256 + IN
63 :

```

Podobně lze upravit program i pro počítače PMD.

Závěr

Možnosti mikropočítačů dovolují vytvářet měřicí přístroje nového typu a je třeba změnit i filosofii měřicí techniky.

Uvedený způsob měření ukazuje, že inteligentní měřicí přístroje nebude nutné složitě ovládat množstvím ovládacích prvků, které zvláště u učitelek a učitelů na středních i základních školách vzbuzovaly obavy a byl proto potlačován experimentální ráz fyziky i dalších přírodních věd.

Z výukových programů fyziky ve vyspělých zemích, jak je možné vidět z množství nabízených pomůcek i z televizních pořadů pro školy vyplývá, že tento experimentální ráz je třeba do výuky přírodních věd vrátit, aby se zlepšila účinnost výukového procesu.

Rovněž pro amatérskou praxi i pro oprávňenou činnost jsou inteligentní, počítači řízené přístroje velkým přínosem.

na str. 3, 2. sloupec má být místo silové čáry procházejí motorem správně rotořem, str. 4, sl. 1: Čím je větší motor . . . tím není záběrový proud větší, je pouze větší důsledek proudového nárazu, rozběhový proud působí déle,

str. 4, sl. 2: Správně je u motorů pro těžký rozbeh s odporovým vinutím kotvou menší záběrový proud,

str. 13, obr. 21: na hřidle je zcela vpravo kuličkové ložisko a další dva výstupky nejsou vývody vznikajícího napětí, ale přívody pro budoucí vinutí,

str. 13, sl. 3: správně má být tachogenerátor jsou střídavé stroje,

str. 25, obr. 67: pod obrázkem je správně uvedeno, že jde o usměrnění jednocestné, v textu k obrázku je však nesprávně uvedeno, že jde o usměrnění jednofázové,

str. 26, obr. 78: kontakt K2/1 je správně kreslen v klidu jako rozpojený,

str. 26, sl. 3: v textu je nesprávně použit termín pracovní nulový vodič. Podle ČSN má být správně „střední volič“.

V příštím čísle
ELEKTRONICKÁ
KUCHAŘKA II
Výběr zajímavých
a praktických zapojení

JEDNOČIPOVÉ MIKROPOČÍTAČE

Ing. M. Reznák

Mikrokontroléry (jak se říká jednočipovým mikropočítačům) jsou vlastně mikroprocesory, které jsou svým uspořádáním navrženy speciálně pro monitorování a řízení různých mechanismů a procesorů, spíše než pro běžnou manipulaci s daty. Mikrokontroléry vždy obsahují tzv. časovače (různé typy), které umožňují synchronizaci s vnějším okolím – s reálným světem. Systémy takto navržené se nazývají řídící systémy v reálném čase (real-time-control-system).

Složitost i výkonnost těchto IO se stále zvětšuje. Tento trend je možné sledovat u mikrokontrolérů fy Intel řady 8048, fy Motorola 6801 a fy Intel typu 8051, které byly od roku 1976 uváděny na trh přibližně v intervalech 2,5 roku. 8048 má jeden časovač 8bitový, 6801 má jeden časovač 16bitový a obvod 8051 má dva 16bitové časovače. Mikrokontrolér nové generace od fy Intel, označovaný 8096, má nový rychlý subsystém V/V (High-Speed I/O), který umožňuje činnost 4 až 8 časovačům.

Aby bylo možné sledovat vývoj a vlastnosti jednotlivých generací mikrokontrolérů fy Intel, je nutné popsat jejich charakteristické vlastnosti:

1. generace – 1976 – řada MCS-48

Typy mikrokontrolérů – 8048, 8748
8049, 8749
8050, 8750
8021
8022

8bitová CPU
1/24 Kbyte ROM
64/128/256 bytů RAM
1 čítač/časovač
paralelní V/V
8bitový převodník A/D

2. generace – 1980 – řada MCS-51

Typy mikrokontrolérů – 8051, 8751
8052, 8752

8bitová CPU
4/8 Kbyte ROM
128/256 bytů RAM
2 čítače/časovače
paralelní V/V
sériový V/V

3. generace – 1983 – řada MCS-96

Typy mikrokontrolérů – 8394, 8794
8395, 8795
8396, 8796
8397, 8797

16bitová CPU
8 Kbytů ROM
232 bytů RAM
čítače/časovače
paralelní V/V
sériové V/V
rychlé V/V
10bitový převodník A/D
hlídací časovač
pulsní šířkově modulovaný výstup

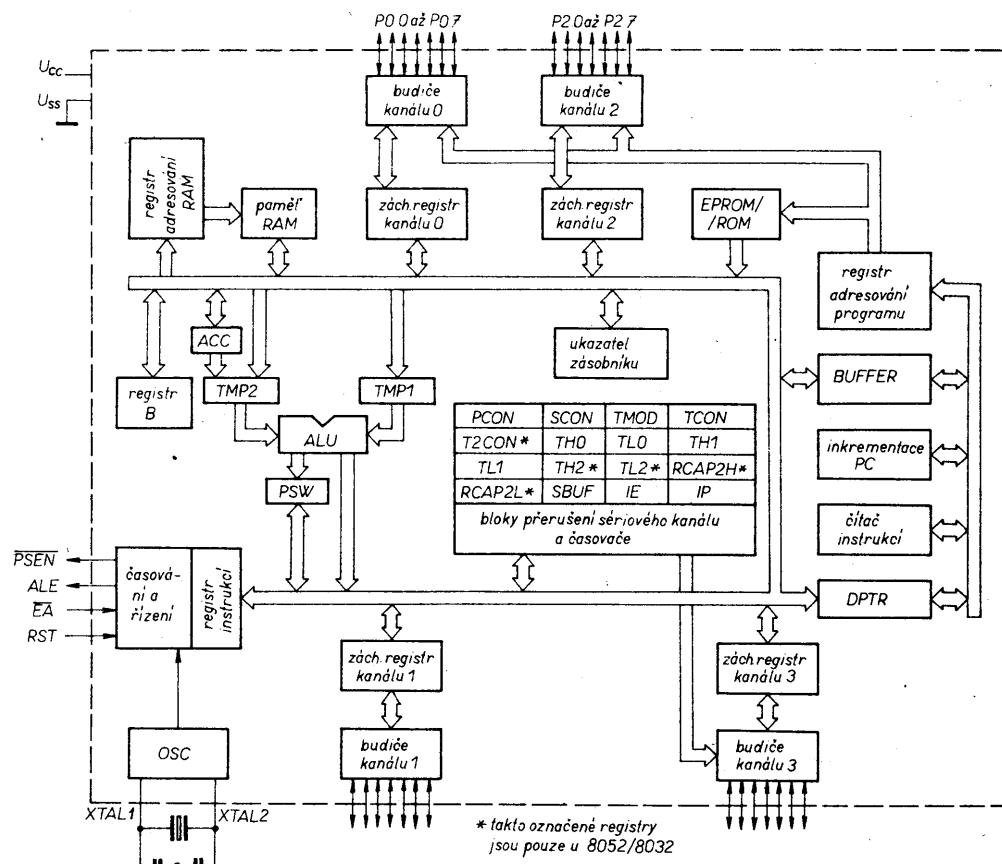
Tab. 1. Obvody řady 8051/8052

Druh	Technologie výroby	Paměť programu na čipu	Paměť dat na čipu
8051AH	HMOS II	4 K ROM	128
8031AH	HMOS II	nemá	128
8751H	HMOS I	4 K EPROM	128
80C51	CHMOS	4 K ROM	128
80C31	CHMOS	nemá	128
8052	HMOS II	8 K ROM	256
8032	HMOS II	nemá	256

a další 16bitový čítač/časovač, který může být použit jako čítač, časovač nebo ke generování „baudových“ rychlostí pro sériový kanál. Ve funkci čítač/časovač může pracovat buď v režimu „16bitový s automatickým přednastavením“ nebo „16bitový záchytň“. Tyto nové vlastnosti jsou popsány dále. Vývody uvedených typů jsou popsány individuálně u datových listů.

Základní vlastnosti mikropočítačů řady MCS-51 jsou:

– 8bitová základní jednotka CPU,



Obr. 1. Blokové schéma architektury obvodů řady 8051/8052

- obvod oscilátoru a časování,
- 32 linek vstup/výstup
- 64 Kbytů vnější paměti dat,
- 64 Kbytů vnější paměti programu,
- dva 16bitové čítače/časovače,
- 5 zdrojů přerušení,
- zcela duplexní sériový kanál,
- Booleovský procesor.

Organizace paměti

Jednočipový mikropočítač typu 8051 má oddělený adresovací prostor pro paměť programu a paměť dat. Paměť programu (Program Memory) může být až 64 Kbytů, přičemž spodní 4 K bytů mohou být umístěny přímo na čipu. Paměť dat (Data Memory) může být také až 64 Kbytů. Tato paměť je umístěna mimo čip kromě 128 bytů (které jsou na čipu) a 20 tzv. speciálních funkčních registrů (Special Function Registers). Označení, druh registru a adresa registru jsou v tab. 2.

Tab. 2. Přehled speciálních funkčních registrů

*ACC	Akumulátor		0E0H				
*B	Registr B		0E0H				
*PSW	Stavové slovo programu		0D0H				
SP	Ukazatel zásobníku		81H				
DPTR	Ukazatel dat		83H				
			a 82H				
*PO	Kanál 0		80H				
*P1	Kanál 1		90H				
*P2	Kanál 2		0A0H				
*P3	Kanál 3		0B0H				
*IP	Řízení priority		0B8H				
*IE	Řízení povolení		0A8H				
	řízení poverení						
TMOD	Řízení režimu		89H				
	čítače/časovačů						
*TCON	Řízení		88H				
	čítače/časovačů						
TH0	Čítač/časovač 0	(vyšší byte)	8CH				
TL0	Čítač/časovač 0	(nižší byte)	8AH				
TH1	Čítač/časovač 1	(vyšší byte)	8DH				
TL1	Čítač/časovač 1	(nižší byte)	8BH				
*SCON	Řízení sériového		98H				
	kanálu						
SBUF	Buffler pro		99H				
	sériová data						
PCON	Řízení napájení		87H				

U obvodů 8052 jsou navíc tyto speciální funkční registry:

Řízení			
čítače/časovače 2	T2CON	0C8H	
Čítač/časovač 2	TH2	0CDH	
(horní byte)			
Čítač/časovač 2	TL2	0CCH	
(spodní byte)			
Záhytný			
registr čítače/			
/časovače 2			
(horní byte)	RCAP2H	0CBH	
Záhytný			
registr čítače/			
/časovače 2			
(spodní byte)	RCAP2L	0CAH	

Speciální funkční registry (SFRs), označené *, jsou adresovatelné jak bitově, tak i bytově.

Stručný popis těchto registrů

Akumulátor (Accumulator)

Akumulátor ACC je registr s různorodým použitím. V instrukcích se pro něj používá označení A.

Registr B

Registr B se používá během operací násobení a dělení. Jinak může být použit jako další obecný registr.

PSW registr stavového slova programu (Program Status Word)

Stavové slovo programu obsahuje informaci o stavu programu (obr. 2).

Obr. 2. PSW – registr stavového slova programu (Program Status Word Register) (nejvyšší bit 7) (nejnižší bit 0)

CY	AC	F0	RS1	RS0	OV	-	P

CY – Příznak přenosu (Carry flag) – pozice PSW.7. Je nastavován/nulován díky hardware nebo software během několika aritmetických a logických instrukcí.

AC – Příznak pomocného přenosu (Auxiliary Carry flag). Je nastavován/nulován díky hardware během sčítání nebo odčítání i indikaci přenosu nebo záporného přenosu ze 3. bitu – – pozice PSW.6.

F0 – Příznak 0 (Flag 0) – pozice PSW.5. Je nastavován/nulován testován díky software jako uživatelem definovaný stavový příznak.

RS1, popř. RS0 – Řídicí bity pro výběr banky registrů, pozice PSW.4, popř. pozice PSW.3. Nastavovány/nulovány díky software, jak je určeno.

Způsob výběru banky:

RS1	RS0	Banka	Adresy
0	0	Banka 0	(00H až 07H)
0	1	Banka 1	(08H až 0FH)
1	0	Banka 2	(10H až 17H)
1	1	Banka 3	(18H až 1FH)

OV – Příznak přeplnění (Overflow flag) – pozice PSW.2. Je nastavován/nulován díky hardware během aritmetických instrukcí k indikaci přeplnění.

P – Příznak parity (Parity flag) – pozice PSW.0. Je nastavován/nulován díky hardware v každém instrukčním cyklu. Tento bit indikuje sudou parity obsahu akumulátoru, tzn. počet jedničkových bitů.

Ukazatel zásobníku SP (Stack Pointer)

Ukazatel zásobníku je 8bitový registr. Je inkrementován předtím, než se uloží data při provádění instrukcí PUSH a CALL. Po nulování (reset) je SP nastaven na adresu 07H, což znamená, že zásobník začíná na adrese 08H. Jinak může být vlastní zásobník umístěn kdekoliv v rezidentní paměti RAM.

Ukazatel dat (Data Pointer) DPTR

Základní funkci ukazatele dat je uchovat 16bitovou adresu. Skládá se proto z horního bytu (DPH) a spodního bytu (DLP). S ukazatelem dat můžeme pracovat jako s 16bitovým registrem nebo dvěma nezávislými 8bitovými registry.

Kanály 0 až 3 (Ports 0, 1, 2, 3)

Speciální funkční registry P0, P1, P2 a P3 jsou vlastně záhytnými registry kanálů 0 až 3.

Buffer sériových dat (Serial Date Buffer)

Buffer sériových dat se skládá ze dvou oddělených registrů; z vysílačního bufferu (trans-

mit buffer) a příjmacího bufferu (receive buffer). Když jsou data zapsána do SBUF, jsou umístěna do vysílačního bufferu, kde jsou uchována pro pozdější sériový přenos. (Předání bytu dat do SBUF inicializuje sériový přenos). Jestliže jsou data přenášena z SBUF, vychází z příjmacího bufferu.

Registry časovače

Páry registrů (TH0, TL0) a (TH1, TL1) jsou 16bitové čítací registry pro čítače/časovače 0 a 1.

Řídicí registry

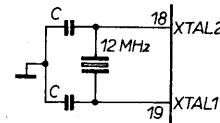
Speciální funkční registry IP, IE, TMOD, TCON a PCON obsahují řídicí a stavové bity pro systém přerušení, pro čítače/časovače a pro sériový kanál.

Záhytné registry

Registrový pár (RCAP2H, RCAP2L) jsou záhytné registry pro čítače 2 v záhytném režimu (capture). V tomto režimu, jako odesva na změnu na vstupu T2EX obvodu 8052, jsou obsahy TH2 a TL2 kopírovány do RCAP2H a RCAP2L. Čítač 2 má také 16bitový „řežim s přednastavením“ a potom si RCAP2H a RCAP2L udržují přednastavené hodnoty tohoto režimu. Více o vlastnostech čítače 2 je dále.

Oscilátor a obvod hodin

X₁ a X₂ jsou vstup a výstup jednostavového invertoru, jenž je umístěn na čipu a může být spolu s vnějšími prvky uspořádán jako Piercův oscilátor (obr. 3).



Obr. 3. Oscilátor obvodů řady 8051/8052 (kapacita kondenzátorů jsou 30 pF ± 10 pF pro krystal a 40 pF ± 10 pF pro keramický filtr)

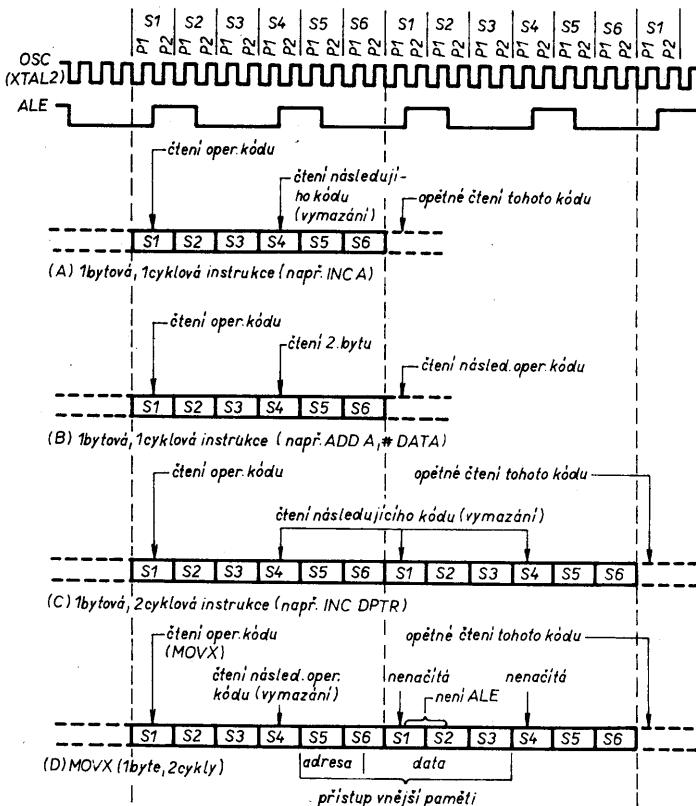
Oscilátor vždy řídí vnitřní generátor hodin, který čípu poskytuje vnitřní hodinové signály. Tyto vnitřní hodinové signály mají oproti oscilátoru poloviční kmitočet a definují vnitřní fáze, stavu a strojní cykly.

Časování CPU

Strojní cyklus se skládá ze šesti stavů (S1 a S6), což je 12 period oscilátoru. Každý stav je rozdělen na poloviční fázi P1 a fázi P2. Strojní cyklus se tedy skládá z dvanácti period oscilátoru, které jsou očíslované od S1P1 (stav 1, fáze 1) až do S6P2 (stav 6, fáze 2). Každá fáze trvá jednu periodu oscilátoru a každý stav dvě periody. Aritmetické a logické operace jsou prováděny během fáze 1 a vnitřní přesuny z registru do registru během fáze 2.

Diagram na obr. 4 ukazuje časování pro načítání/provádění (fetch/execute) instrukcí vzhledem k vnitřním stavům a fázím. Vnitřní hodinové signály nejsou přístupné uživateli, a proto zde signály X2 a ALE slouží pro informaci o průběhu cyklu. Signál ALE je obvykle aktivován dvakrát během každého strojního cyklu: poprvé během S1P2 a S2P1 a podruhé při S4P2 a S5P1.

Provádění jednocyklové instrukce začíná v S1P2, kdy je operační kód zachycen v instrukčním registru. Jedná-li se o dvoubity instrukci, druhý byte je načítán během S4 téhož strojního cyklu. V případě jednobity instrukce se objevuje ještě jedno načítání (fetch) v době S4, avšak načtený byte (který může být následující opkód) je ignorován a čítač programu není inkrementován.



Obr. 4. Zobrazení průběhu načítání/provádění instrukcí

V každém případě je provedení instrukce kompletní na konci strojního cyklu – tzn. ve stavu S6P2.

— Na obr. 4 je časování pro 1bytové a 2bytové instrukce, které trvají jeden strojní cyklus. Většina instrukcí jednočipového mikropočítače typu 8051 trvá jeden cyklus. Pouze instrukce MUL a DIV potřebují ke svému provedení dobu delší než dva strojní cykly – a to dokonce 4 cykly.

Obyčejně jsou dva byty operačního kódu načteny z paměti programu během každého strojního cyklu. Výjimkou je pouze instrukce MOVX, což je 1bytová instrukce trvající 2 cykly, která má přístup do vnější paměti dat. Během provádění instrukce MOVX jsou dvě načítání přeskočena, zatímco je vnější datová paměť adresována a strobována. V dolní části obr. 4 je znázorněno časování pro obvyklé 1bytové 2cyklové instrukce a pro instrukci MOVX.

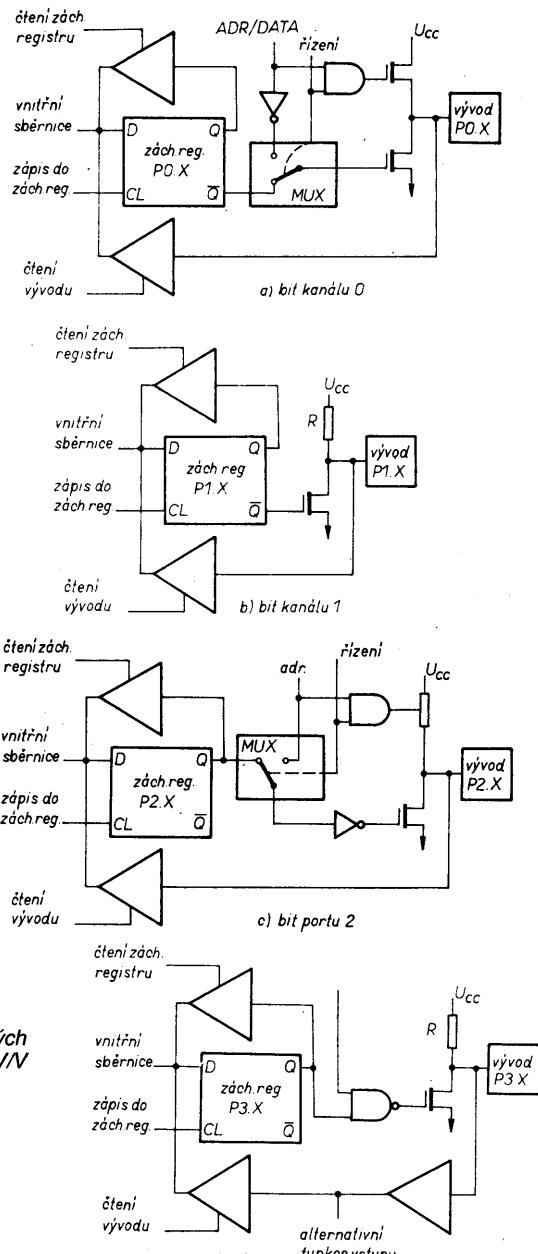
Struktura kanálů V/V a jejich funkce

Všechny čtyři kanály mikropočítače typu 8051 jsou obousměrné. Každý kanál se skládá ze záhytného registru (speciální funkční registry P0 + P3), výstupního budiče a vstupního bufferu (input buffer).

Výstupní budiče (output drivers) kanálů 0 a 2 a vstupní buffery kanálu 0 se používají ke zprostředkování přístupu do vnější paměti dat. V tomto případě je na kanálu 0 spodní byte adresy vnější paměti časové multiplexován bytem, který je právě čten nebo zapisován. Je-li adresa 16bitová, kanál 2 obsahuje horní byte adresy vnější paměti.

Jinak vývody (pins) kanálu 2 určují obsah speciálního funkčního registru P2.

Všechny vývody kanálu 3 mají několik funkcí, jak je zřejmé z tab. 3 (pro obvod 8052 jsou to ještě 2 vývody kanálu 1, které jsou v tabulce označeny*).



Obr. 5. Zapojení záhytných registrů a budičů kanálů V/V obvodu 8051

Tab. 3. Vývody 3. kanálu (1. kanálu) s několika funkcemi

Vývod	Další funkce
P3.0	RXD (vstup sériového kanálu)
P3.1	TXD (výstup sériového kanálu)
P3.2	INT0 (vnější přerušení)
P3.3	INT1 (vnější přerušení)
P3.4	T0 (vnější vstup čítače/časovače 0)
P3.5	T1 (vnější vstup čítače/časovače 1)
P3.6	WR (strobování pro zápis do vnější paměti dat)
P3.7	RD (strobování pro zápis do vnější paměti dat)
*P1.0	T2 (externí vstup čítače/časovače 2)
*P1.1	T2EX (přepínač čítače/časovače 2)

Různé funkce mohou být aktivovány pouze tehdy, má-li příslušný bit v příslušném speciálním funkčním registru úrovně 1. V opačném případě je vývod kanálu na úrovni „0“.

přístupu obsah registru P2SFR zůstává nezměněn, avšak do registru P0SFR jsou zapsány samé „jedničky“. Z obr. 5 je dále zřejmé, že jestliže záhytný registr kanálu P3 obsahuje 1, potom je výstupní úroveň řízena signálem označeným „další výstupní funkce“. Skutečná úroveň kanálu P3.X je vždy dostupná volitelné vstupní funkci.

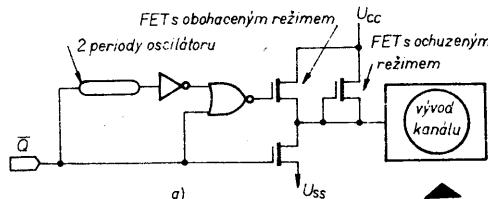
Kanály 1, 2 a 3 mají vnitřní „zvyšovací“ rezistor R (pull-up). Kanál 0 má výstupy s otevřeným kolektorem (open-drain outputs). Každá linka V/V může být použita nezávisle buď jako vstup nebo jako výstup. Kanály 0 a 2 nelze použít jako linky V/V, jsou-li již používány jako adresová/datová sběrnice, ADDR/DATA bus. Chceme-li je použít jako vstup, záhytný registr bitu kanálu (port bit latch) musí mít úroveň 1, která vypíná výstupní tranzistor FET. Pak se (u kanálů 1, 2 a 3) napětí na vývodu zvětší vnitřním „zvyšovacím“ rezistorem, může však být zmenšeno vnějším zdrojem (v tomto případě dodává vývod do zátěže proud – viz katalogový údaj). Díky této vlastnosti se kanály 1, 2 a 3 často označují jako kvazi-obousměrné. Po nulování (reset) se do záhytných registrů všech kanálů zapiší jedničky. Jestliže byly do záhytného registru následně zapsány nuly, pak, chceme-li kanál použít jako vstup, musíme do něj znova zapsat jedničky.

Kanál 0 se do ostatních odlišuje tím, že nemá vnitřní zvyšovací rezistor. Horní tranzistor FET ve výstupním budiči P0 je odpojen ve všech případech kromě toho, kdy kanál používáme jako adresovou/datovou sběrnici pro přístup do vnější paměti, a navíc je zapsána úroveň 1. V důsledku toho záhytný registr kanálu 0 způsobí vypnutí obou tranzistorů FET a tím tzv. „plovoucí úroveň“ vývodu. Za této podmínky může být kanál 0 použit jako vstup s velkou impedancí (je-li zátěž na úrovni 0, nedodává do ní proud). Proto se také kanálu 0 říká plně obousměrný.

Zápis na kanál

Při provádění instrukcí, které mění hodnotu záhytného registru kanálu, je nová hodnota předávána do tohoto registru ve stavu S6P2 posledního cyklu instrukce. Avšak záhytné registry jsou ve skutečnosti vzorkovány svými výstupními buffery pouze během fáze P1 každé periody hodin. (Během fáze 2 udržují výstupní buffery hodnotu, kterou načetly během fáze 1). Nová hodnota se v záhytném registru na vývodu neobjeví až do příští fáze 1, což bude S1P1 příštího strojního cyklu.

Jestliže změna hodnoty v záhytném registru vyžaduje přechod z „0“ do „1“ na kanálech 1, 2 nebo 3, pak se přídavný zvyšovací rezistor sepne během stavu S1P1 a S1P2 toho cyklu, ve kterém nastává přechod. Tímto způsobem lze změnu zrychlit.



Vnější zvyšovací rezistor může dodávat až 100× větší proud, než běžné zvyšovací rezistory. Je však třeba poznat, že vnitřní zvyšovací rezistory jsou vlastně tranzistory řízené polem a nikoli lineární odpory. Uspořádání těchto rezistorů – tranzistorů je znázorněno na obr. 6.

Ve verzji HMOS tvoří pevnou část zvyšovacího „odporu“ FET v depletičním režimu s řídící elektrodou připojenou ke zdroji. Tranzistor umožní dodávat přes vývod proud až 0,25 mA, ovšem je-li „tvrdě“ spojen se zemí. Paralelně s tímto pevným „odporu“ je umístěn další tranzistor (v obohaceném režimu), který je aktivován ve stavu S1, když úroveň na kanálu přechází z „0“ na „1“. Jestliže je během tohoto intervalu vývod kanálu „tvrdě“ spojen se zemí, tento tranzistor umožní dodávat na vývod předavých 30 mA.

Ve verzji CHMOS se zvyšovací „odpor“ skládá ze tří tranzistorů FE s kanálem n. Jen na vysvětlenou: FET s kanálem n (nFET) je sepnut, je-li na jeho řídící elektrodě připojena úroveň log. 1 a vypnut, je-li na ní připojena úroveň log. 0. FET s kanálem p (pFET) pracuje naopak: je sepnut, je-li na řídící elektrodě úroveň log. 0 a rozpojen, je-li na ní úroveň log. 1.

Na obr. 6b je tranzistor, který je sepnut na dobu dvou period oscilátoru po zjištění přechodu z 0 na 1 v záhytném registru. Pokud je sepnut, sepnese se i tranzistor T₃ (malý zvyšovací odpor) prostřednictvím invertoru T₂. Tento invertor a tranzistor formují záhytný registr, který udržuje hodnotu „1“.

Je nutné povšimnout si stavu, kdy je na vývodu úroveň log. 1. Záporný impuls z externího zdroje na tomto vývodu může totiž vypnout T₃, což způsobí, že vývod přechází do plovoucího stavu. T₂ má velmi malý zvyšovací odpor, a je sepnut vždy, když nFET T₄ nevede (tradiční způsob CHMOS). Jeho funkcí je nahradit úroveň „1“ na vývodu tehdy, že na vývodu byla a „ztratila se“ záporným impulsem na vývodu (tzv. glitch).

Zatížení kanálu i připojení

Každý z výstupních budičů (buffer) kanálů 1, 2 a 3 může být zatížen až 4 vstupy LS TTL. Zvnějšku mohou být tyto kanály jako vstupy ve verzji HMOS řízeny obvyklým způsobem některým z obvodů TTL nebo NMOS. Obě verze (HMOS i CHMOS) mohou být řízeny tranzistory s výstupem s otevřeným kolektorem (bipolární) nebo s otevřenou řídící elektrodou (MOS), avšak přechod ze stavu 0 do stavu 1 nebude rychlý. V provedení HMOS, je-li vývod řízen výstupem s otevřeným kolektorem, přechod z 0 do 1 bude muset řídit

FET s poměrně malým odporem v ochuzeném (depletičním) režimu (viz obr. 6a). V provedení CHMOS úroveň 0 připojená na vstup vypíná zvyšovací „odpor“ T₃ a přechod z 0 na 1 bude řízen pouze velmi malým zvyšovacím odporem T₂.

Výstupní budiče kanálu 0 mohou být zatíženy až 8 vstupů LS TTL. Pro napájení vstupů NMOS jsou však nezbytné externí zvyšovací rezistory mimo případ, kdy jsou použity jako adresová/datová sběrnice.

Načtení – změna – zápis na kanál

Jsou dva způsoby, jak přečíst hodnotu na kanálu: instrukce čte buď hodnotu na záhytném registru nebo na vývodu. Instrukce, které upřednostňují čtení hodnoty na záhytném registru před čtením na vývodu jsou ty instrukce, které hodnotu načítou, mohou ji změnit a pak ji znova zapíši do záhytného registru. Tyto instrukce nazýváme „čti-zaměň-zapiš“ a jsou popsány v tab. 4.

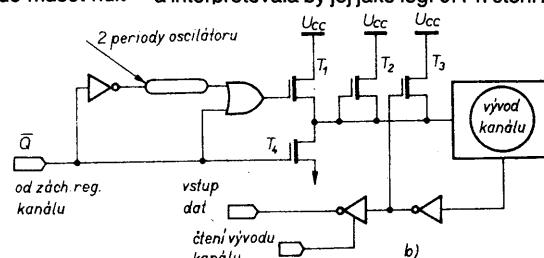
Je-li cílovým operandem kanál nebo bit kanálu, tyto instrukce čtou záhytný registr dříve než vývod.

Tab. 4. Přehled instrukcí typu „čti-zaměň-zapiš“

ANL	(logický AND, např. ANL P1,A)
ORL	(logický OR, např. ORL P2,A)
XRL	(logický EX-OR, např. XRL P3, A)
JBC	(skok, jestliže bit = 1 a vymazání bitu např. JBC P1.1, návěští)
CPL	(doplň bitu, např. CPL P3,0)
INC	(inkrement, např. INC P2)
DEC	(decrement, např. DEC P2)
DJNZ	(decrement a skok, když obsah není nula, např. DJNZ P3, návěští), (přesuň C do bitu Y kanálu X)
MOVX PX.Y, C	(vymaž bit Y kanálu X)
CLR PX.Y	(nastav bit Y kanálu X)
SET PX.Y	

Není očividné, že poslední tři instrukce tohoto seznamu by byly instrukce typu „čti-zaměň-zapiš“, avšak přesto jsou. Čtou byte kanálu, všech 8 bitů, změní adresový bit a zpět celý nový byte zapiší do registru. Důvodem, proč se instrukce tohoto typu obracejí dříve na záhytný registr než na vývod je to, že se takto lze vyhnout možné nedefinovatelné úrovni vývodu. Např. bit kanálu může řídit přímo bázi tranzistoru. Jestliže je do bitu zapsána úroveň log. 1, tranzistor je otevřen.

Kdyby CPU četla ten samý bit kanálu na vývod dříve než na záhytném registru, načítala by vlastně bázové napětí tranzistoru a interpretovala by jej jako log. 0. Při čtení ze



Obr. 6a. Zapojení vnitřních zvyšovacích rezistorů u kanálu 1 a 3 ve verzji HMOS. Tranzistor pracující v obohaceném režimu je sepnut na dobu dvou period oscilátoru po zjištění přechodu z 1 na 0 na vstupu Q

Obr. 6b. Totéž jako na obr. 6a pro verzi CHMOS. Po zjištění přechodu z 1 na 0 na vstupu Q je pFET T₁ sepnut na dobu dvou period oscilátoru. Během této doby je přes invertor sepnut také pFET T₃, čímž je formován záhytný registr, který drží „1“. T₂ je též sepnut

záchranného registru však dostaneme správou úroveň, log. 1.

Druhy přístupu do vnější paměti

Přístup do vnější paměti je dvou typů: přístup do externí paměti programu a přístup do externí paměti dat. Při přístupu do externí paměti programu se používá signál PSEN jako strobovací signál čtení a při přístupu do externí paměti dat to jsou signály RD nebo WR ke strobování paměti (RD a WR jsou vícefunkční vývody 3, kanálu).

Načítání z vnější paměti programu vždy používá 16bitové adresy. Komunikujeme-li s vnější pamětí dat, můžeme používat buď 16bitovou adresu (MOVX, DPTR) nebo 8bitovou adresu (MOVX, RI).

Kdykoliv použijeme 16bitovou adresu, vyšší byte adresy přejde na kanál 2, kde je uchován po dobu trvání čtecího nebo zapisovacího cyklu. Je nutno si povšimnout, že budeč 2. kanálu během celé doby, co jsou vynulovány jedničkové adresové bity, používají velké zvyšovací odpory (je to v průběhu provádění instrukce MOCX a DPTR). Během této doby záchranný registr 2. kanálu (speciální funkční registr) nemusí obsahovat jedničky, a přesto obsah kanálu 2 speciálního funkčního registru není modifikován.

Jestliže cyklus přístupu do vnější paměti bezprostředně následuje za jiným cyklem přístupu do vnější paměti, nezměněný obsah kanálu 2 speciálního funkčního registru se znovu objeví v příštém cyklu.

Jestliže právě používáme 8bitovou adresu (MOVX a RI), obsah kanálu 2 (speciálního funkčního registru) se zachová na vývodech kanálu 2 po dobu provádění cyklu vnější paměti, což zjednoduší stránkování.

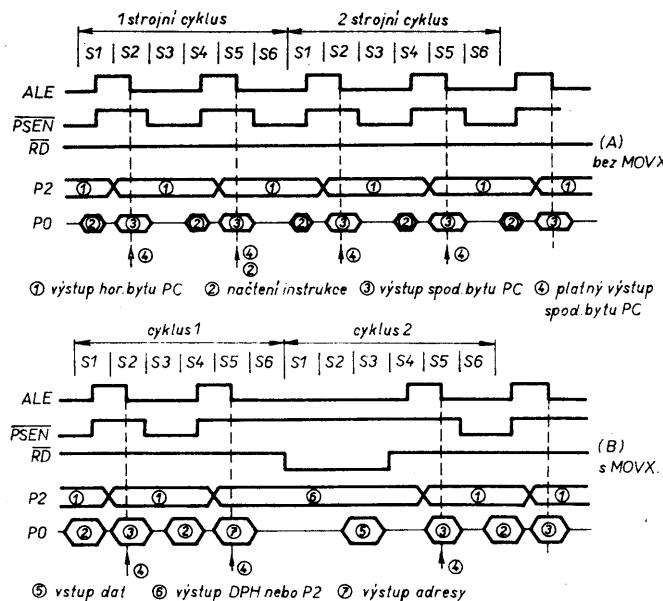
V každém případě je spodní byte adresy časově multiplexován s bytem dat na kanálu 0, signál ADDR/DATA řídí oba tranzistory FET výstupních budičů kanálu 0. Pak vývody kanálu 0 nebudou výstupy s otevřeným kolktorem, a proto se nevyžaduje použít vnější zvyšovací odpory. Signál ALE by mohl být použit k zachycení (capture) bytu adres ve vnějším záchranném registru. Byte adresy je platný při sestupné hraně signálu ALE (address latch enable). Pak se během zapisovacího cyklu objeví byte dat, který má být zapsán na kanálu 0 právě před aktivací signálu WR a zůstane tam, dokud signál WR nebude deaktivován. V cyklu čtení je vstupující byte přejímán na kanál 0 právě před deaktivací strobovacího signálu pro čtení (RD).

Během každého přístupu do vnější paměti zapiše CPU na záchranný registr kanálu 0 (SFR) hodnotu OFFH, což způsobí vymazání jakékoli informace na kanálu 0 speciálního funkčního registru. Vnější paměť programu je přístupná za dvou podmínek:

1. jakmile je signál EA aktivní, nebo
2. jakmile čitač instrukcí (PC) obsahuje hodnotu větší než 0FFFH (1FFFH pro obvod 8052).

Proti mají verze bez paměti ROM signál EA připojený na nízkou úroveň, což umožňuje přístup do spodních 4 Kbytů (8 Kbytů pro 8032) programu a slouží k vyvolávání z vnější paměti.

Když CPU načítá z vnější paměti programu všech 8 bitů kanálu 2, je vyhrazeno pro výstupní funkce a nemůže být tedy použita jako obecný vstup/výstup. Tyto byty obsahují během načítání vnějšího programu horní byte obsahuje čitače instrukcí (PC). Bě-



Obr. 7. Průběhy načítání z externí programové paměti

hem této doby byty kanálu 2, které jsou jedničkové, používají velké zvyšovací odpory.

PSEN

Čtecím strobovacím signálem načítání z vnější paměti je signál PSEN. Signál PSEN není určen pro vnitřní načítání. Během přístupu CPU do vnější paměti programu je signál PSEN aktivován dvakrát během každého cyklu (kromě cyklů při provádění instrukce MOVX), bez ohledu na to, zda načítaný byte je právě potřebný pro průběh instrukce. U aktivovaného signálu PSEN není časování stejně jako u RD. Úplný cyklus signálu RD, včetně aktivace a deaktivace signálu ALE a RD, trvá 12 period oscilátoru. Úplný cyklus signálu PSEN, včetně aktivace a deaktivace signálu ALE a PSEN trvá 6 period oscilátoru. Srovnání těchto cyklů je na obr. 7.

ALE (ADDRESS LATCH ENABLE)

Hlavní funkcí signálu ALE je poskytnout vhodný časový signál, který by zachytil spodní byte adresy na kanálu 0 do vnějšího záchranného registru na dobu načítání programu z vnější paměti. Za tímto účelem je ALE aktivováno dvakrát během každého strojního cyklu. Aktivace však nastává i tehdy, nejedná-li se o načítání z vnější paměti. První impuls signálu ALE v druhém cyklu schází (viz obr. 7). Proto je v systému, který nepoužívá vnější paměť dat signál ALE, aktivován konstantní rychlosť 1/6 kmitočtu oscilátoru a může být použit pro vnější časování.

Překrývání vnější paměti programu a dat

V některých aplikacích se vyžaduje, aby byl realizován program ze stejné fyzické paměti, která je právě používána k uchování dat. V mikropočítači typu 8051 může být prostor vnější paměti dat a programu spojen použitím logického součinu AND u signálu PSEN a RD. Kladný logický součin těchto dvou signálů vyprodukuje strobovací signál aktivní při úrovni log. 0 pro čtení, který se může použít pro spojenou fyzickou paměť. Protože je cyklus PSEN rychlejší než cyklus signálů RD, vnější paměť musí být dostačně rychlá, aby se přizpůsobila cyklu PSEN.

Čítače/časovače (TIMER/COUNTERS)

Mikropočítač typu 8051 má dva 16bitové čítače/časovače, čítač/časovač 0 a čítač/časovač 1. Každý může být uspořádán buď jako čítač nebo jako časovač. Obvod 8052 má navíc další čítač/časovač 2.

Popis funkce a architektura

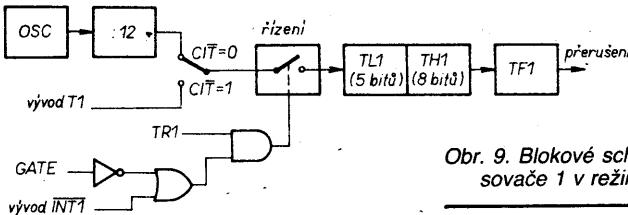
Ve funkci „časovače“ je registr inkrementován během každého strojního cyklu. Tuto funkci můžeme tedy brát jako načítání strojních cyklů. Protože se strojní cyklus skladá z dvanácti period oscilátoru, je rychlosť načítání rovna 1/12 kmitočtu oscilátoru.

Ve funkci „čítače“ je registr inkrementován po přechodu úrovně 1 → 0 na příslušném vnějším vývodu pro vstup, tzn. T0 nebo T1. Vnější vstup je vzorkován ve stavu S5P2 každého strojního cyklu. Když vzorkování ukazuje hodnotu 1 v jednom cyklu a hodnotu 0 v druhém, načítání (count) je inkrementováno. Tato nová načtená hodnota se objeví v registru ve stavu S3P1 následujícího cyklu. Protože tato činnost trvá dva cykly, maximální rychlosť načítání je 1/24 kmitočtu oscilátoru. Na střídání vnějšího signálu nejsou žádná omezení, avšak aby bylo zaručeno, že obdržená hodnota bude vzorkovaná alespoň jednou než se změní, měla by být podržena přinejmenším jeden celý strojní cyklus.

Mimo rozdělení na „čítač“ nebo „časovač“ má každý čítač/časovač čtyři operační režimy. Režimy 0, 1 a 2 jsou stejné pro oba, režim 3 je odlišný. Čítač/časovač 2 v obvodu 8052 má tři pracovní režimy: záchranný, s automatickým znovuzaváděním a jako generátor přenosové rychlosti.

Čítač/časovač 0 a čítač/časovač 1

Čítače/časovače 0 a 1 jsou obsaženy v obou obvodech, 8051 i 8052. Funkce čítače nebo časovače je zvolena řídicímitem C/T ve speciálním funkčním registru, označeném TMOD. Tyto dva čítače/časovače mají čtyři operační režimy zvolené bity M1



a M0 ve speciálním funkčním registru TMOD (viz obr. 8).

Obr. 8. TMOD : registr řízení režimů čítače/časovače

Nejvyšší bit

Nejnižší bit

	GATE	C/T	M1	M0	GATE	C/T	M1	M0
časovač 1				časovač 0				
GATE je-li nastaven, probíhá řízení hradlem. Čítač/časovač x je uvolněn pouze tehdy, má-li vývod INTx velkou úroveň a TRx = 1. Je-li vynulován, čítač je uvolněn, jestliže je řídicí bit TRx nastaven.				Nejvyšší bit				
C/T	Vybírá činnost : „čítače“ nebo „časovače“.				Nejnižší bit			
	C/T = 0 činnost „časovače“ (vstup z vnitřního systému hodin),							
	C/T = 1 činnost „čítače“ (vstup vývodu Tx).							
M1	Definování režimu řízení čítače/časovače.							
M0	M1 M0 způsob provozu							
0 0	čítač jako u obvodu 8048. TLx slouží jako 5bitová dělička.							
0 1	16bitový čítač/časovač. THx a TLx jsou zapojeny do kaskády, není zde předřazena žádná dělička.							
1 0	8bitový čítač/časovač s automatickým přednastavením. THx obsahuje hodnotu, která je po každém přetečení přenesena do TLx (přednastavení).							
1 1	(časovač 0) TL0 je 8bitový časovač, čítač je řízen standardními bity časovače 0, TH0 je 8bitový časovač, který je řízen pouze řídicími bity časovače 1.							
1 1	(časovač 1) Čítač/časovač 1 je zastaven.							

REŽIM 0

Nastavení čítače/časovače 0 (nebo 1) do režimu 0 vypadá stejně jako u 8048AH, což je 8bitový čítač s předřazenou děličkou 32. Obr. 9 ukazuje činnost režimu 0 v aplikaci na čítač/časovač 1.

V tomto režimu je časovací registr uspořádán jako 13bitový registr. Jakmile hodnota v registru přechází ze samých jedniček na samé nuly, nastaví se příznak přetečení časovače TF1. Čítací vstup je připojen do čítače/časovače, když TR = 1 a bud' GATE = 0 nebo INT1 = 1. (Nastavení GATE = 1 dovoluje, aby byl časovač řízen vnějším vstupem INT1, což zjednoduší

Obr. 9. Blokové schéma činnosti čítače/časovače 1 v režimu U: 13bitový čítač

Obr. 10. TCON : registr řízení čítače/časovače

Nejvyšší bit

Nejnižší bit

Pozice	Sym- bol	Funkce a význam
TCON.7	TF1	Příznak přetečení časovače 1.0.
TCON.5	TF0	Nastavený pomocí hardware při přetečení. Nulováný hardwarem (přivolání obslužného podprogramu).
TCON.6	TR1	Řídicí bity chodu časovače 1.0.
TCON.4	TR0	Nastavený/nulováný softwarem pro zapnutí/vypnutí čítače/časovače.
TCON.3	IE1	Příznaky změny úrovně vnějšího přerušení. Nastavený pomocí hardware, jakmile je detekována hrana vnějšího přerušení 1, 0. Vynulován, když obsluha přerušení probíhá.
TCON.1	IE0	
TCON.2	IT1	Řídicí bity pro určení druhu vnějšího přerušení.
TCON.0	IT0	Nastavený/nulováný programově pro specifikování bud' vnějšího přerušení hranou/vnějšího přerušení úrovní („0“).

měření šířky impulsu.) TR1 je řídicí bit ve spec. funkčním registru TCON, GATE je řídicí bit ve SFR TMOD (viz obr. 10). 13bitový registr se skládá z 8 bitů registru TH1 a nížších 5 bitů registru TL1. Horní 3 bity registru TL1 jsou neurčeny a mohou být zanedbány. Nastavení (setting) příznaku chodu časovače 1 (TR1) nezpůsobí vymazání registru.

Činnost režimu 0 je stejná pro čítač/časovač 0 i 1. Pro srovnání dosaďme TR0, TF0 a INT0 místo signálů čítače/časovače 1 (obr. 9). Rozlišujeme však dva bity GATE – jeden pro čítač/časovač 1 (TMOD.7) a jeden pro čítač/časovač 0 (TMOD.3).

REŽIM 1

Režim je stejný s režimem 0 kromě toho, že registr časovače pracuje na 16 bitech.

REŽIM 2

V režimu 2 je registr časovače uspořádán jako 8bitový čítač (TL1) s automatickým zpětným přednastavením (automatic reload), jak ukazuje obr. 11. Přetečení registru TL1 nejen že nastaví příznak TF1, ale zpětně naplní registr TL1 obsahem registru TH1 (který je předem nastavený pomocí softwa-

re). Zpětné naplnění neovlivní registr TH1.

Činnost režimu 2 je stejná i pro čítač/časovač 0.

REŽIM 3

Čítač/časovač 1 si v režimu udržuje svou hodnotu (nemění se). Účinek je stejný jako při nastavení TR1 = 0.

Čítač/časovač 0 v režimu 3 zřizuje registry TL0 a TH0 jako dva oddělené čítače. Logické schéma je na obr. 12. Registr TL0 používá tyto řídicí bity čítače/časovače 0: C/T, GATE, TR0, INT0 a TF0. Registr TH0 má funkci časovače (načítání strojních cyklů) a používá TR1 a TF1 časovače 1. Tudíž TH0 řídí přerušení od časovače 1. Tento režim provozu je speciálně navržen pro aplikace, které vyžadují zvláštní 8bitový čítač nebo časovač. Dá se tedy říci, že, je-li časovač 0 v režimu 3, má obvod 8051 čítače/časovače tří a obvod 8052 potom čítače/časovače čtyři. Během tohoto režimu může být časovač 1 použit buď jako:

- generátor přenosové rychlosti pro sériový kanál,
- v nějaké aplikaci, která nevyžaduje přerušení,
- může se spouštět a zastavovat tím způsobem, že je uváděn do tohoto režimu a přepínán do režimu jiného.

Časovač 2

Tento časovač obsahuje pouze obvody 8052. Stejně jako časovače/čítače 0 a 1, může tento časovač pracovat buď jako čítač, nebo jako časovač. Funkce je vybrána bitem C/T2 ve speciálním funkčním registru T2CON. Má 3 režimy provozu (obr. 13):

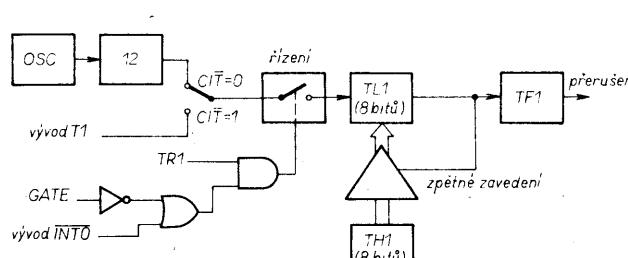
- záhytný (capture),
- s automatickým přednastavením (autoreload),
- jako generátor přenosové rychlosti.

Tyto režimy jsou zvoleny nastavením řídicích bitů ve speciálním funkčním registru T2CON (tab. 5).

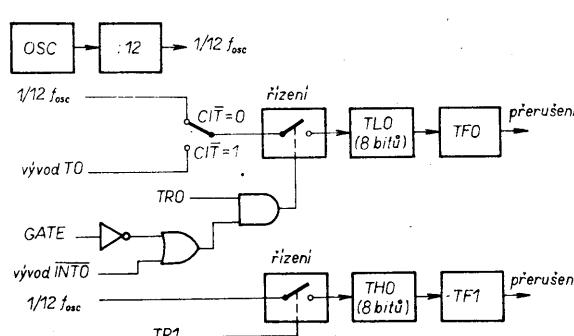
Tab. 5. Nastavení provozních režimů 2. časovače

RCLK + TCLK	CP/RL2	TR2	režim
0	0	1	16bitový s automatickým přednastavením
0	1	1	16bitový záhytný generátor přenosové rychlosti vypnut
X	X	0	

V záhytném režimu jsou možné dva způsoby provozu, které se dají nastavit bitem EXEN2 ve speciálním funkčním registru T2CON.



Obr. 11. Blokové schéma činnosti čítače/časovače 1 v režimu 2: 8bitové automatické přednastavení



Obr. 12. Blokové schéma činnosti čítače/časovače v režimu 0: dva 8bitové čítače

Obr. 13. T2CON: Řídicí registr čítače/časovače 2
Nejvyšší bit

TF2	EXF2	RCKL	TCLK	EXEN2	TR2	C/T2	CP/RL2
-----	------	------	------	-------	-----	------	--------

Pozice	Symbol	Funkce
T2CON.7	TF2	Příznak přetečení časovače 2. Nastavován při přetečení časovače 2, nulován musí být programově. Tento příznak nebude nastaven, je-li nastaven jeden z bitů RCLK = 12 nebo TCLK = 1.
T2CON.6	EXF2	Vnější příznak časovače 2, který musí být nulován programově. Nastaven je za předpokladu, že EXEN2 = 1 a na vývodu T2EX je indikována sestupná hrana, která spouští přednastavení nebo přepis okamžitých hodnot registrů (zachycení).
T2CON.5	RCLK	Příznak hodin přijímání. Je-li tento příznak nastaven, používá potom sériový kanál pro své vnitřní přijímací hodiny v režimech 1 a 3 impulzy z přetečení časovače 2. Je-li tento příznak vynulován, potom je pro vnitřní přijímací hodiny využíván přetečení časovače 1.
T2CON.4	TCLK	Příznak hodin vysílání. Je-li tento příznak nastaven, používá potom sériový kanál pro své vysílací hodiny v režimech 1 a 3 impulzy z přetečení časovače 2. Je-li tento příznak vynulován, potom je pro vnitřní vysílací hodiny využíván přetečení časovače 1.
T2CON.3	EXEN2	Vnější uvolňovací příznak časovače 2. Jestliže je tento příznak nastaven a přitom není časovač 2 použit pro časování sériového kanálu, umožňuje tento bit přednastavení nebo přepis okamžitých hodnot registrů (zachycení) jako důsledek na sestupnou hrana signálu na vývodu T2EX. Jestliže je EXEN2 = 0, časovač 2 nereaguje na změny na vývodu T2EX.
T2CON.2	TR2	Příznak pro spouštění (zastavování) časovače 2. Je-li TR2 = 1, časovač 2 je spuštěn.
T2CON.1	C/T2	Výběr režimu činnosti.
T2CON.0	CP/RL2	Je-li C/T2 = 0 – časovač 2 je ve funkci časovače (čítá s kmitočtem $f_{osc}/12$). Je-li C/T2 = 1, pracuje ve funkci čítače vnějších událostí (spouštěn sestupnou hranou). Příznak zachycení (přednastavení). Jestliže je tento příznak nastaven a navíc je EKEN2 = 1, potom se v důsledku sestupné hrany na vývodu T2EX přepisují registry časovače 2 do záhytných registrů (zachycení). Jestliže je tento příznak vynulován a navíc je EKEN = 1, potom se přednastavují registry časovače 2 jako důsledek buď sestupné hrany na vývodu T2EX nebo v důsledku přetečení časovače 2. Za předpokladu, že jeden z bitů RCLK a TCLK je nastaven, potom je tento příznak CP/RL2 ignorován a časovač je po každém přetečení přednastaven.

Je-li bit EXEN2 = 0, časovač 2 pracuje jako 16bitový čítač nebo časovač, který po přetečení nastaví příznak TF2. Tento příznak může být potom použit ke generování přerušení.

Je-li bit EXEN2 = 1, časovač 2 pracuje stejně jako v předešlém případě, avšak s jednou funkcí navíc. Sestupná hrana impulsu (přechod ze stavu 1 do 0) na externím vývodu T2EX způsobí, že okamžité hodnoty registrů časovače 2 (TL2 a TH2) jsou zapsá-

ny do speciálních funkčních registrů RCAP2L a RCAP2H. Dále ještě sestupná hrana impulsu způsobí nastavení bitu.

EXF2 (obsažen ve speciálním funkčním registru T2CON), který může podobně jako příznak přetečení TF2 generovat přerušení. Tento záhytný režim je znázorněn na obr. 14.

V režimu s automatickým přednastavením jsou také možné dva způsoby provozu, které jsou volitelným bitem EXEN2 ve speciálním funkčním registru T2CON.

Je-li bit EXEN2 = 0, potom přetečení časovače 2 nezpůsobí pouze nastavení příznaku přetečení TF2, ale způsobí zpětné naplnění časovače 2 (TL2 a TH2) hodnotami vregistrech RCAP2L a RCAP2H. Tyto registry RCAP2L a RCAP2H jsou již předem programově naplněny hodnotami přednastavení.

Je-li EXEN2 = 1, časovač 2 pracuje stejně jako v předešlém případě, avšak má funkci navíc. Sestupná hrana impulsu (přechod ze stavu 1 do 0) na externím vývodu T2EX způsobí, že okamžité hodnoty registrů časovače 2 TL2 a TH2 jsou přepsány do speciálních funkčních registrů RCAP2H. Taktéž tato hrana způsobí nastavení bitu EXF2 v registru T2CON), který může podobně jako příznak přetečení TF2 generovat přerušení. (Tato přídavná vlastnost je stejně jako přídavná vlastnost v záhytném režimu.)

Činnost za tohoto režimu je zřejmá z obr. 15.

Režim jako generátor přenosové rychlosti se nastaví nastavením bitů RCLK a TCLK (RCLK = 1 a TCLK = 1). Podrobně bude popsán v článku o sériovém kanálu.

Řízení čítačů/časovačů a stavové registry

Speciální funkční registry TMOD a TCON se používají k definování režimů a řídicích funkcí čítačů/časovačů. Jakmile instrukce změní obsah TMOD, TCON, změna je zachycena v SFR a ovlivňuje funkce ve stavu

S1P1 prvního cyklu příští instrukce. Registry jsou na obr. 8 a 10.

Sériový interface

Sériový kanál je zcela duplexní (tzn. přenáší a přijímá zároveň). Může také začít přijímat druhý byte předtím, než byl dříve přijatý byte přečten z přijímacího registru. (Jestliže však první byte není kompletní, jeden z bytů se ztratí). Vysílací a přijímací registry sériového kanálu jsou přístupné v oblasti speciálních funkčních registrů SBUF. Zápis do registru SBUF naplňuje vysílací registr, avšak čtení SBUF přebírá obsah jiného, fyzicky odděleného registru, a to registru přijímacího.

SCON je SFR, který se používá k určení operačního režimu pro sériový kanál. Přijímá 9. bit dat (RB8 – bude popsáno dále) a obsahuje další návěští stavů.

Sériový kanál může pracovat ve čtyřech režimech. Tyto čtyři režimy popíšeme nejdříve ve stručnosti a dále se jim budeme věnovat podrobněji.

REŽIM 0

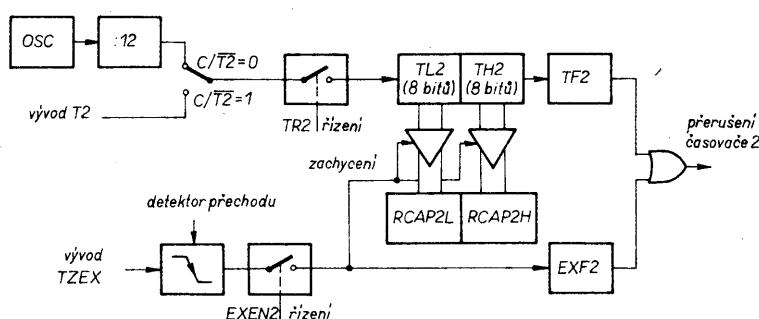
Sériová data jsou přenášena vývodem RxD (jak vstup, tak výstup), vývodom TxD vystupují přenosové hodinové impulsy, které mají pevný kmitočet rovný $1/12$ kmitočtu oscilátoru. V tomto režimu je přenášeno 8 bitů dat, přičemž nejvyšší bit je přenášen jako první.

REŽIM 1

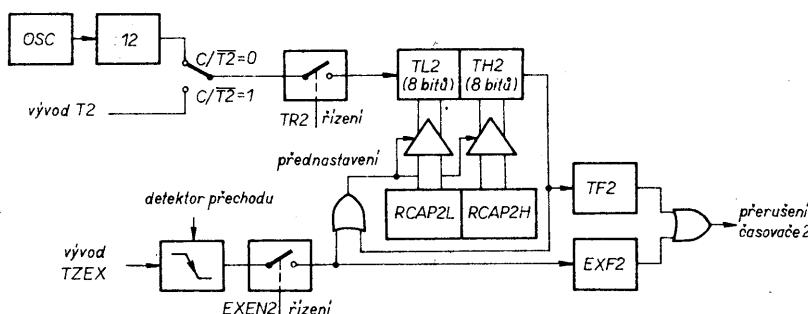
V tomto režimu je přenášeno 10 bitů, které jsou vysílány vývodem TxD. Skládají se ze startovacího bitu (logická hodnota 0), dále následuje 8 bitů dat (nejvyšší bit je přenášen jako první) a nakonec je přenesen stop-bit (logická hodnota „1“). Po příjmu dat je stop-bit přenesen na pozici RB8 ve speciálním funkčním registru SCON. Přenosová rychlosť je v tomto režimu proměnná.

REŽIM 2

Při tomto způsobu provozu je přenášeno 11 bitů. Vysílány jsou vývodem TxD, přijímá-



Obr. 14. Blokové schéma činnosti časovače 2 v záhytném režimu



Obr. 15. Blokové schéma činnosti časovače 2 v režimu automatické přednastavení

ny vývodem RxD. Skládají se ze startovacího bitu (úroveň log. 0), dále následuje 8 bitů dat (nejnižší bit je přinášen jako první), dále je 9. programovatelný bit a na konci zase stop-bit (úroveň log. 1).

Při vysílání může být 9. bitu přiřazena úroveň log. 0 nebo 1 programově (jedná se o bit TB8 ve speciálním funkčním registru SCON, do kterého může např. být přenesena hodnota paritního bitu P z PSW). Při příjmu je hodnota 9. bitu přenesena na pozici RB8 ve speciálním funkčním registru SCON. Stop-bit je při příjmu ignorován. Přenosová rychlosť je programovatelná a může být buď 1/3 nebo 1/64 kmitočtu oscilátoru.

REŽIM 3

Tento režim je ve všem shodný s režimem 2, až na přenosovou rychlosť. Ta je v tomto případě proměnná.

Ve všech čtyřech režimech je vysíláni inicializováno instrukcí, která používá SBUF jako cílový registr. Příjem je v režimu 0 inicializován nastavením bitů v SCON (RI = 0 a REN = 1. V ostatních režimech začíná příjem startovacím bitem (REN musí být nastaven na 1).

Multiprocesorová komunikace

Režimy 2 a 3 umožňují speciálním způsobem multiprocesorovou komunikaci. V těchto režimech je přijato 9 bitů dat, přičemž 9. bit přechází do RB8, pak přichází stop-bit. Sériový kanál může být naprogramován tak, že jakmile je stop-bit přijat, bude přerušení sériového kanálu aktivováno pouze tehdy, jestliže RB8 = 1. To je umožněno nastavením bitu SM2 v registru SCON.

Použití: Když chce hlavní procesor přenést blok dat do jednoho z několika podřízených procesorů, nejprve vyšle byte adresy, který identifikuje podřízený procesor. Byte adresy se odlišuje od bytu dat devátým bitem (9. bit bytu adresy = „1“, zatímco 9. bit bytu dat = „0“). Nastavení SM2 = „1“ umožňuje, že žádný podřízený procesor nebude přerušen bytem dat, avšak bytem adresy budou přerušeny všechny. Pak každý podřízený procesor může tento byte adresy přijmout a rozlišit, zda byl určen jemu. Adresovaný podřízený procesor pak vymaže svůj bit SM2 a připraví se na přijetí bytu dat, které přijdou. Podřízené procesory, které nebyly adresovány, nechají byty SM2 nastavené na „1“, jsou zaměstnány svou prací a ignorují přicházející data.

SM2 nemá použití v režimu 0, v režimu 1 může být použit pro kontrolu platnosti stop bitu. Při příjmu v režimu 1, je-li SM2 = 1, nebude aktivováno přerušení, dokud nebude přijat platný stop bit.

Řídící registr sériového kanálu

Řízení a stavy sériového kanálu jsou obsaženy v řídicím registru sériového kanálu SCON na obr. 16. Tento registr neobsahuje pouze bity pro řízení režimů, ale také 9. bit pro vysílání a příjem (TB8 a RB8) a bity přerušení od sériového kanálu (TI a RI).

Baudové rychlosti

„Baudová“ rychlosť režimu 0 je daná: $1/12$ kmitočtu oscilátoru. Baudová rychlosť režimu 2 závisí na bitu SMOD v SFR registru PCON. Jestliže je SMOD = „0“, čož je po počátečním nulování, baudová rychlosť je $1/64$ kmitočtu oscilátoru, je-li SMOD = „1“, baudová rychlosť je $1/32$ kmitočtu oscilátoru.

Obr. 16. SCON: Registr řízení sériového kanálu

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SM0							specifikují režimy sériového kanálu
SM1							SM0 SM1 REZIM Popis Přenosová rychlost
	0	0	0		posuvný fosc/12		registr
	0	1	1		8bitový proměnná		UART
	1	0	1		9bitový fosc/64		UART nebo/32
	1	1	3		9bitový proměnná		UART
SM2				umožňuje multiprocesorovou komunikaci			
				v režimech 2 a 3. V režimu 2 nebo 3, je-li:			
				SM2 = „1“, pak RI nebude aktivováno,			
				jestliže je přijat současně 9. bit (RB8 = „0“).			
				Je-li SM2 = „1“ v režimu 1, RI			
				pak nebude aktivováno, jestliže nebyl			
				přijat platný stop-bit.			
REN				umožňuje sériový příjem. Nastaven softwarem. REN = „1“ umožňuje přijetí,			
				REN = 0 blokuje příjem.			
TB8				je 9. bit dat, který bude v režimech			
				2 a 3 přenášen. Nast./nul. softwarem,			
				jak je požadováno.			
RB8				v režimech 2 a 3 to je 9. bit dat, který byl			
				přijat. V režimu 1, jestliže SM2 = „0“,			
				RB8 je stop-bitem, který byl přijat. V režimu 0 se nepoužívá.			
TI				příznak přerušení vysílání. Nastavení			
				pomocí hardware na konci 8. doby bitu			
				v režimu 0 nebo na začátku doby stop-bitu v ostatních režimech při každém			
				sériovém vysílání. Musí být nulován			
				softwarem.			
RI				příznak přerušení příjmu. Nastavení			
				hardwarem na konci 8. doby bitu v režimu 0 nebo v půlce doby stop-bitu v ostatních režimech při každém sériovém příjmu (mimo viz SM2). Musí být vynulován			
				programově.			

přednastavení – vyšší čtverice bitů registru TMOD = 0010B) je baudová rychlosť:

$$\text{baud. rychlos} \text{t} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{kmitočet oscilátoru}}{12 \times (256 - \text{TH1})}$$

Dosáhnout velmi malé baudové rychlosti lze tak, že povolíme přerušení časovače 1 a nastavíme ho, aby pracoval jako 16bitový časovač (vyšší čtveřice bitů TMOD = 0001B).

Použití časovače 2 ke generování přenosových rychlostí

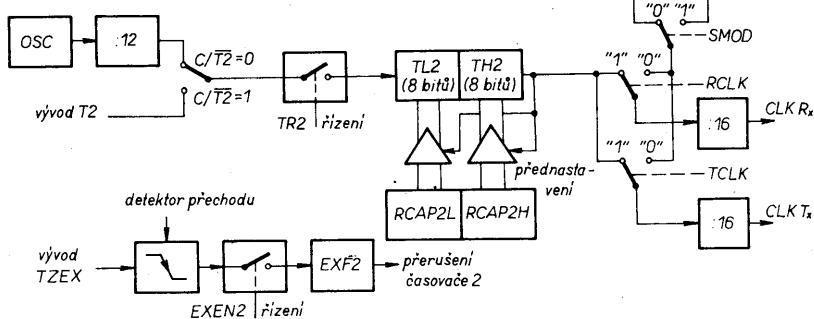
Časovač 2 je v obvodu 8052 ve funkcií generátoru přenosové rychlosti, je-li ve speciálním funkčním registru T2CON nastaven bit TCLK a (nebo) bit RCLK. Z toho je vidět, že přenosové rychlosti vysílání a příjmu mohou být rozdílné. Funkce časovače 2 v tomto režimu je znázorněna na obr. 17. Činnost je dosti podobná režimu s automatickým přednastavením. Přetečení v registru TH2 totiž způsobí, že obsahy registrů RCAP2H a RCAP2L (které jsou předem nastaveny programově) jsou automaticky přeneseny do registrů časovače 2 (přednastavení).

Tab. 6. Různé „baudové“ rychlosti a možný způsob jejich získání pomocí časovače 1

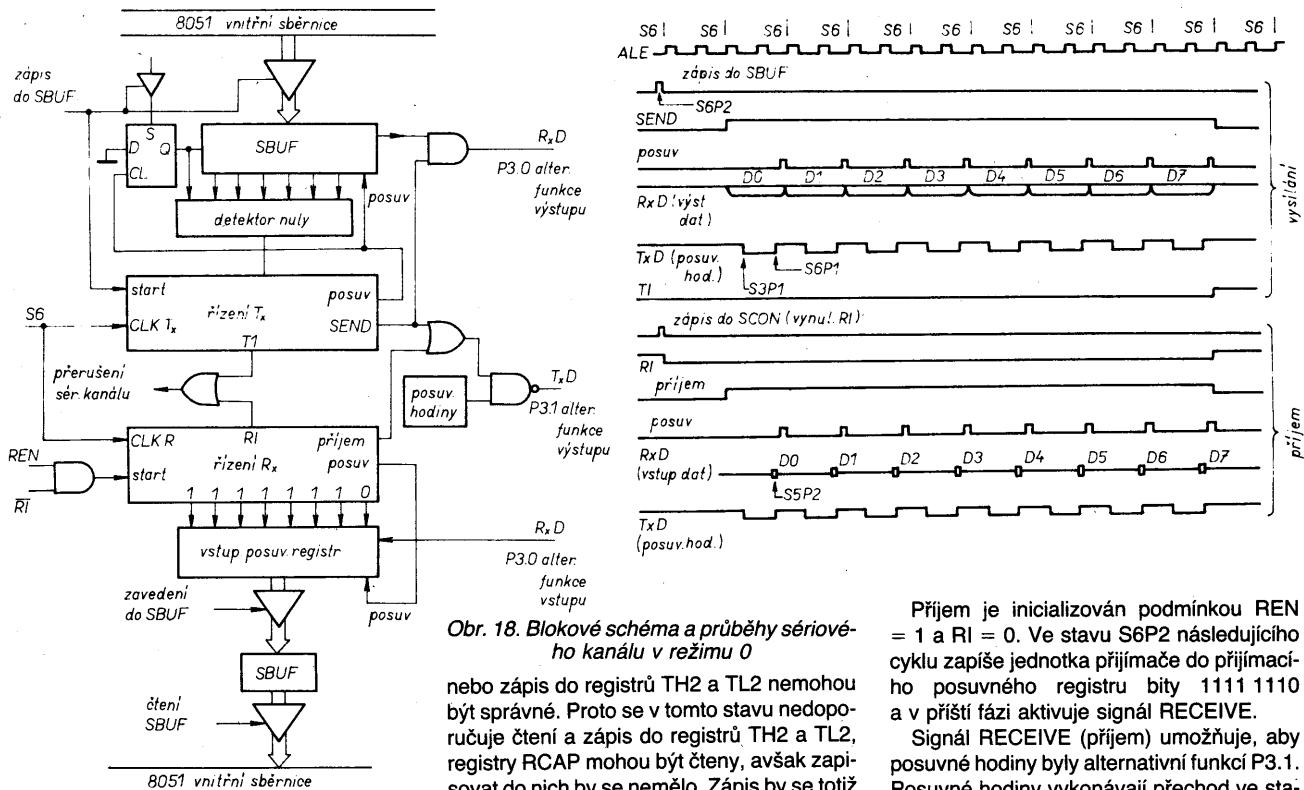
Baud. rychlosť	Kmitočet oscilátoru [MHz]	Časovač 1			
		SMOD	C/T	režim	pred- nasta- vení
režim 0	-				
max:					
1 MHz	12	X	X	X	X
režim 2					
max:					
375 k	12	1	X	X	X
režimy					
1,3: 62,5 k	12	1	0	2	FFH
19,2 k	11,059	1	0	2	FDH
9,6 k	11,059	0	9	2	FDH
4,8 k	11,059	0	0	2	FAH
2,4 k	11,059	0	0	2	FAH
1,2 k	11,059	0	0	2	E8H
137,5	11,986	0	0	2	1DH
110	6	0	0	2	72H
110	12	0	0	1	FEE8H

$$\text{baud. rychlos} \text{t} = \frac{2^{\text{SMOD}}}{32} \times \text{rychlos} \text{t} \text{ pretecen} \text{i} \text{ casova} \text{c} \text{e} 1$$

Přerušení čítače/časovače 1 by v tomto případě nemělo být povoleno. Čítač/časovač může být uspořádán pro provádění činnosti „časovače“ nebo „čítače“ v některém z režimů. V aplikacích (např. uspořádání pro činnost časovače v režimu automatického)



Obr. 17. Blokové schéma činnosti časovače 2 v režimu generátoru přenosových rychlostí (CLK RX – hodinový signál příjímače)



Obr. 18. Blokové schéma a průběhy sériového kanálu v režimu 0

nebo zápis do registrů TH2 a TL2 nemohou být správné. Proto se v tomto stavu nedoporučuje čtení a zápis do registrů TH2 a TL2, registry RCAP mohou být čteny, avšak zapisovat do nich se nemělo. Zápis se totiž mohl překrývat s přednastavením a mohl by způsobit chybu (zápisu nebo přednastavení). Je-li zápis z nějakých důvodů potřebný, je nutno nejdříve časovač zastavit (vynutit TR2) a pak pracovat s registry RCAP nebo TH2 a TL2.

REŽIM 0

Sériová data vstupují a vystupují prostřednictvím vývodu Rx.D. Z Tx.D vystupují „posuvné“ hodinové impulsy (shift clock). Je přenášeno 8 bitů dat (nejnižší bit je přenášen jako první). Baudová rychlosť je pevná – 1/12 kmitočtu oscilátoru.

Na obr. 18 je zjednodušený funkční diagram sériového kanálu v režimu 0 a příslušné časování.

Přenos je inicializován jakoukoli instrukcí, která používá SBUF jako cílový registr. Signál „zapiš do SBUF“ zavede ve stavu S6P2 „1“ do místa devátého bitu „vysílačského posuvného registru“ a naznačí tím řídicímu bloku vysílače, aby začal s přenosem. Vnitřní časování je takové, že mezi signálem „zapiš do SBUF“ a aktivací signálu SEND uplyne jeden úplný strojový cyklus.

Signál SEND umožňuje, aby výstup z posuvného registru byl alternativní funkci kanálu P3.0 a aby posuvné hodiny byly alternativní funkci kanálu P3.1. Posuvné hodiny mají nízkou úroveň během stavů S3, S4 a S5 každého strojového cyklu a vysokou během stavu S6, S1 a S2. Ve stavu S6P2 každého strojového cyklu, ve kterém je příkaz SEND aktivní, je obsah vysílačského posuvného registru posuván o jedno místo doprava.

Protože jsou bity dat posuvány doprava, zleva je vysílačský registr plněn nulami. Jakmile je nejvyšší bit v pozici výstupu z posuvného registru, potom „1“, která byla původně nastavena v deváté pozici, je hned vlevo od nejvyššího datového bitu a všechny ostatní pozice vlevo odtud obsahují samé nuly. Ten to stav naznačuje bloku řízení vysílače, aby vykonal poslední posuv, a pak deaktivuje signál SEND a nastaví T1. Obě tyto činnosti se provedou ve stavu S1P1 desátého strojového cyklu (počítáno od signálu „zapiš do SBUF“).

Příjem je inicializován podmírkou REN = 1 a RI = 0. Ve stavu S6P2 následujícího cyklu zapiše jednotka přijímače do přijímacího posuvného registru bity 1111 1110 a v příští fázi aktivuje signál RECEIVE.

Signál RECEIVE (příjem) umožňuje, aby posuvné hodiny byly alternativní funkci P3.1. Posuvné hodiny vykonávají přechod ve stavech S3P2 a S6P1 každého strojového cyklu. Ve stavu S6P2 každého strojového cyklu, ve kterém je signál RECEIVE aktivní, je obsah přijímacího posuvného registru posouván o jedno místo doleva. Hodnota, která je doplňována zprava, je hodnota, která byla vzorkována na vývodu P3.0 ve stavu S5P2 stejněho strojového cyklu. Protože bity vstupují zprava, jedničky vystupují ven zleva.

Jakmile „nula“, která byla původně nastavena do pravého krajního místa posuvného registru, bude v levém krajním místě, naznačí bloku řízení přijímače skončení posledního posuvu a naplnění SBUF. Ve stavu S1P1 desátého strojového cyklu (počítáno od signálu „zapiš do SCON“, který vymaže RI) je RECEIVE deaktivováno a RI nastaveno.

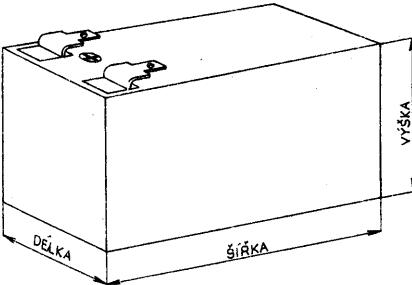
REŽIM 1

V režimu 1 je přenášeno 10 bitů (vysíláno přes TXD a přijímáno přes RXD): jsou to start-bit (log. 0) a stop-bit (log. 1) a 8 bitů dat (nejnižší bit je přenášen jako první). Při příjmu přejde stop-bit do RB8 registru SCON. U mikropočítače typu 8051 je baudová rychlosť určena rychlosťí přeplnění časovače 1. U obvodu 8052 je přenosová rychlosť určena rychlosťí přetečení buď časovače 1, nebo časovače 2, nebo obou (jedna rychlosť je určena pro vysílání, druhá rychlosť je určena pro příjem).

Obr. 19 ukazuje zjednodušený funkční diagram sériového kanálu v režimu 1 a časování pro vysílání a příjem.

Vysílání je inicializováno tou instrukcí, která použije SBUF jako cílový registr. Signál „zapiš do SBUF“ také zavádí „1“ do místa 9. bitu vysílačského posuvného registru a naznačí řídicí jednotce vysílače, že je požadováno vysílání. Přenos ve skutečnosti začne ve stavu S1P1 toho strojového cyklu, který následuje za dalším přetečením předřazené dílčí 16. (Pokračování)

AKUMULÁTORY BEZÚDRŽBOVÉ HERMETICKÉ



vysoká životnost	6 let/200 – 1000 cyklů
malé rozměry	cca 30 typů a provedení
jednoduché dobíjení	možnost sér./paral. řazení
velké vybíjecí proudy	malý vnitřní odpor
rozsah prac. teplot	–60 °C, +60 °C
ekonomie nákladů	výhodný poměr kapacita/cena
snadná manipulace	nehrozí rozlití elektrolytu
robustní konstrukce	
libovolná pracovní poloha	
vysoká spolehlivost	

- Vysoká životnost, spolehlivost a jednoduchost použití čini hermetické akumulátory optimálním zdrojem pro použití v mnoha aplikacích, pro napájení zálohovacích zdrojů a autonomních zařízení.
- Na objednávku u větších množství a pravidelných odběrů nabízíme sortiment akumulátorů v provedení 6 V a 12 V v řadě kapacit od 0,5 Ah/6 V do 80 Ah/12 V.
- Pro podrobnou technickou dokumentaci k akumulátorům i dalším prvkům zabezpečovací techniky a informace o možnosti okamžitého odběru nás laskavě kontaktujte na naší brněnské adresě.

Typ	MC	VOC	rozměry
12 V/1,2 Ah	540,-Kčs	442,-Kčs	97 × 42 × 51 mm
12 V/1,9 Ah	630,-Kčs	485,-Kčs	178 × 34 × 60 mm
12 V/4,2 Ah	740,-Kčs	580,-Kčs	90 × 70 × 100 mm
12 V/6,5 Ah	760,-Kčs	622,-Kčs	151 × 65 × 94 mm

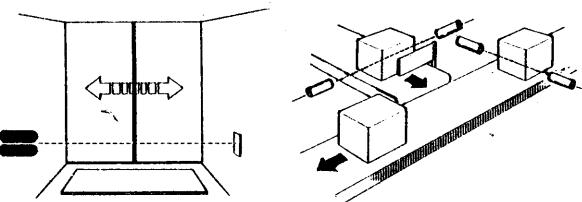
OLYMPIC controls Ltd. – BEZPEČNOSTNÍ SYSTÉMY
Šumavská 31, 612 64 BRNO
tel. (05)7111/344, 368, fax (05) 749150

INFRACERVENÉ SNÍMACE TELCO No1

Naše snímače jsou použitelné všude tam, kde je potřebné detektovat překážku, počítat výrobky, snímat hladinu tekutých i sypkých hmot v zásobnicích, rozlišovat plochu výrobků, chránit pracovní prostor atd.

Výhody našich snímačů jsou:

- velmi malé rozměry, vysoký výkon (dosah až 35 m)
- absolutní odolnost proti znečištění vysílače i přijímače
- vynikající odolnost proti vibracím, rázu, vodě atd.
- žádné problémy se sousošostí senzoru
- velmi kvalitní vyhodnocování jednotky (1 až 8 kanálů)



Technická data

	LT	LT110	LRB	LR	LR110
Dosah	0–15 m	0–35 m	0–7 m	0–15 m	0–35 m
Krytí					IP 67
Povolené vibrace			10–55 Hz s amplitudou 3 mm		
Povolený ráz			30 g		
Prac. teplota			již od –25 °C		
Přívodní kabel		5 nebo 15 m	PVC 2 × 0,25 mm		
Materiál – čočka			Polykarbonát		
– pouzdro			ABS nebo nikl, mosaz		
Výstup			relé 5 A/240 V		
Zivotnost			100 000 hod./25 °C		

Díky témtu vlastnostem je možné využít naše snímače i v těžkých podmínkách těžebních, dřevozpracujících, potravinářských a jiných provozů. Za kvalitu ručí výrobce – dánská firma TELCO. Poskytujeme záruku 14 měsíců. Na požádání Vám obratem zašleme bezplatně informace.

Naše adresa: OLYMPIC CONTROLS Ltd. Šumavská 31, 612 64 Brno
tel. 7111/368, 344 fax 741 427

INZERCE



čitelně, aby se předešlo chybám vznikajícím z nečitelnosti předloh.

PRODEJ

ORWY 6901 (290), SL1452 (580), TDA5660P (170), sat. kon. Masprom – Jap. $F = 1,3$ dB max. (1950), BFR90, 91 (28). F. Krunt, Řepová 554, 190 00 Praha 9, tel. 68 70 870.

Siemens: BFR90, 91, BFQ69, sady po 5 ks (92, 102, 279). Zajistíme Vám dovoz dalších zahraničních součástek a výrobků. Firma ZAVAX, Box 27, 142 00 Praha 411.

Kvalitní málo používaný tranzistorový osciloskop do 250 MHz, profesionálny, vstanován kalibrátor, obrazovka 6 × 10 cm, vyr. ZSSR, typ 1 175. Cena 7000. P. O. Box 5, 914 51 Trenčianské Teplice.

RŮZNÉ

LHOTSKÝ – E. A., electronic actuell nabízí vybrané druhy součástek za výhodné ceny. Nabídkový seznam i s cenami na požádání zdarma zašleme. P. O. Box 40, 432 01 Kadaň 1.

! DJ0QR – GÜNTER HÜTTER !

Koupíme k inkurantním Wehrmacht radiopřístrojům originální příruční knížky všech druhů. Pište na adr. Günter Hüttner, Post. Box 2129, D-8990 Lindau BRD.

VELKOOBCHOD SE SOUČÁSTKAMI PRO ELEKTRONIKU

Vám nabízí široký sortiment

součástek a konstrukčních prvků předních světových výrobců.

Přijďte, píšte, objednávejte, telefonujte.

S.O.S. Electronic spol. s r.o., Loosova 1c, 638 00 Brno, **tel. 06 - 52 40 08**
fax 06 - 52 40 00

• Pryč se zastaralými konstrukcemi •

